UAM
UNIVERSIDAD AUTONOMA
DE MADRID

Departamento de Ingeniería Informática

An Approach for Automatic Generation of on-line Information Systems based on the Integration of Natural Language Processing and Adaptive Hypermedia Techniques

Dissertation written by **Enrique Alfonseca Cubero**
under the supervision of **Pilar Rodríguez Marín**

Madrid, 22nd February 2003

Contents

Abstract xv

Resumen xvii

Acknowledgements xix

I Introduction 1

1 Introduction 3

1.1 Motivations 5

1.1.1 Problem to solve 5

1.1.2 Linguistic Motivation 6

1.1.3 Goals and expected results 7

1.2 General purposes of the research 8

1.3 Architecture: theoretical approach 11

1.3.1 Contents identification (off-line step) 11

1.3.2 Contents generation (on-line step) 13

1.4 The WELKIN implementation: high level design 13

1.4.1 Off-line processing 14

1.4.2 On-line processing 15

1.5 Textual resources used 17

1.5.1 Web sites built 17

1.5.2 Training and test material 18

1.6 Contributions of the thesis 19

1.7 Thesis overview 20

2 Adaptive Hypermedia 23

2.1 Adaptive Hypermedia review 23

2.1.1 Applications of Adaptive Hypermedia Systems 24

2.1.2 Characteristics to which the systems adapt 26

2.1.3 Elements that are adapted 27

2.1.4 Design of the hyperspace 31

2.2 AH with Natural Language Processing Techniques 34

2.2.1 Techniques for creating the Knowledge Base 36
2.2.2 Techniques for planning the hypermedia pages 37
2.2.3 Techniques for generating the text 39
2.3 Summary 42

II Off-line processing: Domain Knowledge Acquisition **43**

3 Lexical Knowledge Acquisition **47**
3.1 Representation of a LKB 47
3.2 Conceptual ontologies 48
3.2.1 Information in an Ontological LKB 49
3.2.2 Applications 50
3.3 Existing conceptual ontologies 52
3.3.1 WordNet 52
3.3.2 Other existing lexical resources 53
3.4 Acquisition of Conceptual Ontologies 55
3.4.1 A classification of Ontology Learning algorithms 56
3.4.2 Ontology Refinement 60
3.5 Summary 63

4 Distributional Semantics **65**
4.1 Introduction 65
4.1.1 A definition of hyponymy and synonymy 66
4.2 The Distributional Semantics model 70
4.2.1 Distance metric between contexts 71
4.2.2 Distance metrics between meanings 76
4.3 Empirical analysis 78
4.4 Summary 80

5 Distributional Semantics Applied to LKA **81**
5.1 Introduction 81
5.2 General Named Entity Identification 82
5.2.1 From NE to GNE 83
5.2.2 Overlapping with word-sense disambiguation 85
5.3 Framework for evaluation 85
5.3.1 Training data 85
5.3.2 Evaluation metrics 86
5.3.3 Distance-based evaluation metrics 87
5.3.4 Test data 90
5.4 An algorithm for classifying unknown terms 91
5.4.1 Identification of terms 94
5.5 Attachment of new synsets 98
5.5.1 Signatures 99

5.5.2 Similarity metric . . . 102
5.5.3 Combining the similarity measures . . . 103
5.6 Experiments and Results . . . 104
5.6.1 Results on weighting the signatures . . . 105
5.6.2 Results on different signatures . . . 106
5.6.3 Comparison with other approaches . . . 106
5.7 Summary and discussion . . . 107

6 The Documental Database **111**
6.1 Linguistic annotation . . . 111
6.2 Term classification with hyponymy patterns . . . 112
6.2.1 Automatic extraction of hyponymy patterns . . . 114
6.2.2 Modifications to the original algorithm . . . 115
6.2.3 Experiments and Results . . . 116
6.3 Identification of time expressions and events . . . 117
6.3.1 A Framework for identifying time expressions . . . 118
6.3.2 Detection of events in a text . . . 119
6.3.3 Anchoring events in time . . . 120
6.4 Identification of domain-dependent terminology . . . 122
6.4.1 A procedure to identify automatically scientific names . . . 123
6.4.2 Results . . . 124
6.5 Summary and discussion . . . 126
6.5.1 Term Classifications . . . 126

III On-line processing: Text generation **129**

7 Automatic Text Summarisation **133**
7.1 Introduction . . . 133
7.2 Text extraction . . . 135
7.2.1 Edmundsonian paradigm . . . 136
7.2.2 Summarisation using discourse analysis . . . 137
7.3 Text Abstraction . . . 139
7.3.1 Abstracting with Information Extraction . . . 140
7.3.2 Abstracting by identifying events . . . 141
7.3.3 Abstracting with internal semantic representations . . . 142
7.4 Multi-document summarisation . . . 144
7.5 Summarisation applied to hypermedia generation . . . 146
7.6 Summary and discussion . . . 147

8 Adaptive Generation of Contents **149**
8.1 User profiles . . . 150
8.1.1 User interests . . . 150
8.1.2 Topic filtering . . . 154

8.1.3 Available time and reading speed . . . 156
8.2 An algorithm for adaptively summarising the text contents . . . 157
8.2.1 Evaluation . . . 162
8.3 User interface . . . 166
8.4 Gathering Information from the Internet . . . 168
8.4.1 Internet Search . . . 172
8.4.2 Relevant-term multi-document coreference resolution . . . 173
8.4.3 Generating the summary . . . 174
8.4.4 Evaluation . . . 175
8.5 Summary and discussion . . . 178

IV Evaluation and Conclusions 181

9 Three case studies and a usage evaluation 185
9.1 Charles Darwin's *A Naturalist's Voyage round the World* . . . 185
9.1.1 Classification of new terms . . . 186
9.1.2 Time expressions detection . . . 187
9.1.3 Scientific names identification . . . 187
9.1.4 Performance of the off-line processing . . . 188
9.1.5 Topic classification . . . 189
9.1.6 Summaries from Internet . . . 189
9.2 William Osler's *The evolution of modern medicine* . . . 193
9.2.1 Classification of new terms . . . 193
9.2.2 Performance results . . . 194
9.2.3 Topics classification . . . 195
9.3 Georg Hegel's *History of philosophy* . . . 196
9.3.1 Classification of new terms . . . 197
9.3.2 Performance results . . . 197
9.3.3 Topics classification . . . 197
9.4 Usage evaluation . . . 198
9.4.1 Experiment . . . 199
9.5 Summary and discussion . . . 204

10 Conclusions and Future Work 205
10.1 Contributions . . . 205
10.2 Comparison with other existing approaches . . . 206
10.3 Future work . . . 209
10.4 Discussion and Future Work of the Components . . . 210
10.4.1 Term Classification . . . 210
10.4.2 Distributional Semantics . . . 212
10.4.3 Analysis of temporal expressions . . . 212
10.4.4 Summarisation . . . 213
10.4.5 User modelling . . . 214

10.4.6 Collection of documents from Internet 214

Postscript **217**

A Abbreviations **219**

B Engineering Work **221**
B.1 Linguistic tools 221
B.1.1 Segmentation 221
B.1.2 Part-of-speech tagger 222
B.1.3 Morphological analyser 222
B.1.4 Chunk parsers 223
B.1.5 Quotes solver 225
B.1.6 Parsing 226
B.2 Changes to the WordNet structure 226
B.2.1 Microtheories 227
B.2.2 Instances and concepts 228

C Example of generated summaries **237**

D Introducción **247**
D.1 Motivación 249
D.1.1 Problema 249
D.1.2 Motivación lingüística 250
D.1.3 Resultados esperados 251
D.2 Objetivos generales de la investigación 252
D.3 Arquitectura: descripción teórica 255
D.3.1 Identificación de contenidos (fuera de línea) 255
D.3.2 Generación de contenidos (en línea) 256
D.4 La implementación de WELKIN: diseño de alto nivel 257
D.4.1 Procesamiento fuera de línea 258
D.4.2 Procesamiento en línea 259
D.5 Recursos de texto utilizados 260
D.5.1 Sitios web construidos 260
D.5.2 Material de entrenamiento y de pruebas 261
D.6 Contribuciones de la tesis 262
D.7 Estructura de la tesis 263

E Conclusiones y trabajo futuro **267**
E.1 Contribuciones 267
E.2 Comparación con otros trabajos 268
E.3 Trabajo futuro 271
E.4 Discusión y trabajo futuro de los componentes 272
E.4.1 Clasificación de terminología 272
E.4.2 Semántica Distribucional 274

E.4.3 Análisis de expresiones temporales 274
E.4.4 Generador de resúmenes 275
E.4.5 Modelado de usuario 276
E.4.6 Recogida de documentos de Internet 276

References **279**

List of Figures

1.1 Overview of the architecture for generating content pages for an adaptive site. 14

2.1 Taxonomy of adaptive hypermedia technologies, from Brusilovsky [2001]. 28
2.2 Different ways to design the hyperspace in an adaptive hypermedia application, from Carro [2001]. 32
2.3 Architecture of an adaptive hypermedia system based on Natural Language Generation techniques, from Milosavljevic et al. [1998]. (a) Traditional NLG architecture; (b) dynamic NLG hypertext system. 35
2.4 Semantic network with concepts, instances, and relations amongst them. 37
2.5 Rules for generating different outputs depending on the user profile and the context [Not et al., 1998]. 40

3.1 Example of entry in a Lexical Knowledge Base, corresponding to the word *comes* 48
3.2 (a) Semantic network of concepts and relations amongst them. (b) The same semantic network, redrawn to make evident that the IS_A relationship establishes a taxonomy. 49
3.3 A small portion of the WordNet taxonomy that is rooted on the synset *entity*. 53

4.1 Example of taxonomy (extracted from WordNet). 76

5.1 Possible taxonomies for the classes in the MUC-7 Named Entity task. 83
5.2 Possible extension of some entities in the MUC-7 NE task with subclasses, using the ontological approach. 83
5.3 An initial taxonomy, a set of new concepts and instances, and the taxonomy extended with the concepts and instances from the set. 84
5.4 Example of taxonomy, an unknown relevant concept u_j, its correct generalisations g_j and the generalisations proposed by three hypothetical algorithms h_{ik}. 86
5.5 Learning accuracy in three different cases. (a) When the proposed concept is correct, but too general. (b) When the proposed concept is incorrect. (c) When there are different ways to compute Learning Accuracy. 88
5.6 Example of taxonomy in which each node is labelled with its Information Content. 89
5.7 Example of sentence annotated. 92
5.8 General architecture of the system that classifies unknown terms in a lexical ontology. 94
5.9 Algorithm for identifying unknown words and proper names from the source documents. . . . 95
5.10 Interpretation of the word Ajax found in *The Iliad*. When it refers to any person called *Ajax*, then it is a concept; while when it refers to a particular person, it is an instance 98

5.11 This shows the WordNet synset *avatar* and its hyponyms. *Rama* should be an instance of *avatar*, but it is also a concept which has three different instances: the three incarnations. . . 98
5.12 Pseudo-code of the algorithm for finding the correct place where the unknown synset u will be attached in the ontology . . . 99
5.13 Algorithm for automatically collecting lists of context words for each WordNet synset, from Internet, from Agirre et al. [2000a]. . . . 100

6.1 Sample of the start of a document that contains a chapter from *The Voyages of the Beagle*, with some annotations for the document structure, syntactic chunks and dates. . . . 112
6.2 Sample of a paragraph from *The Voyages of the Beagle*, with some syntactic annotation. . . . 113
6.3 General architecture of the system. . . . 119
6.4 Pseudo-code of the algorithm for finding temporal relations between events. . . . 121
6.5 Recall-precision curve for identifying the language of the first 200 pairs of words from *The Voyage of the Beagle* and 200 scientific names. . . . 124

7.1 Example cohesion graphs. . . . 138
7.2 Clausal analysis of Mars text [Marcu, 1999]. . . . 140
7.3 RST-analysis of Mars text from Figure 7.2, showing promotion [Marcu, 1997]. The numbers in the rectangles show which children are the nucleus of the relationships. . . . 140
7.4 Text for information extraction, from the Third Message Understanding Conference [Sundheim, 1991]. . . . 141
7.5 Template generated for the text in Figure 7.4 . . . 141
7.6 Concept graph where the two events from sentence (16), *spy* and *carry*, are related through arcs. . . . 142

8.1 Algorithm for generalising the topic signatures using WordNet. . . . 153
8.2 Form with which the user can select one or several predefined stereotypes, or state that he wants to define his own profile. . . . 155
8.3 Form with which an already existing user can log in, or a new user can create a profile. . . . 156
8.4 Reading speed test. The text to be read is sensibly longer than that shown in the image. . . . 158
8.5 Reading comprehension test. There is a total of ten questions about the passage, and the percentage of correct answers is the comprehension rate. . . . 159
8.6 Initial population of summaries. Each line in the figure is the genotype of a summary, which contains the numbers of the sentences that are selected for the summary. . . . 160
8.7 Text that was supplied to the judges to generate a summary. It included a short description of the user interests, the total number of sentences in the text, and the number of sentences that had to be selected. Each sentence was numbered in order to facilitate the annotation. . . 164
8.8 Paragraphs shown to a user interested about history, with a compression rate of 22%. The uncompressed paragraphs are shown in Figure C.1. . . . 166
8.9 Paragraphs shown to a user interested about history, with a compression rate of 33%. The uncompressed paragraphs are shown in Figure C.1. . . . 167
8.10 Paragraphs shown to a user interested about history, with a compression rate of 45%. The uncompressed paragraphs are shown in Figure C.1. . . . 167
8.11 WELKIN main page showing a section of *The Voyages of the Beagle*. . . . 169

8.12 WELKIN main page showing a section of *The Voyages of the Beagle.* 170
8.13 Algorithm for collecting accurate additional information from the Internet. 172
8.14 Generated page about the concept *Valparaiso.* 176
8.15 Paragraphs discarded about the concept *Valparaiso*, which refer to a different city. 177

9.1 Table of contents of *A Naturalist's Voyage round the World* for a user interested in biology (first six chapters). 190
9.2 Table of contents of *A Naturalist's Voyage round the World* for a user interested in geography (first six chapters). 190
9.3 Table of contents of *A Naturalist's Voyage round the World* for a user interested in history (first six chapters). 191
9.4 Results classifying new terms from Osler's *The evolution of modern medicine.* 194
9.5 System Acceptability Attributes (Nielsen [1990], from Fritz [1995]). 198
9.6 Eason's framework of usability (Eason [1993], from Fritz [1995]). 199
9.7 Number of animals found in function of the reading efficiency in English. The top line represents the performance of the users with WELKIN, and the line below the performance of the users with a standard text editor. 203
9.8 Number of biographies found in function of the reading efficiency in English. The top line represents the performance of the users with WELKIN, and the line below the performance of the users with a standard text editor. 204

B.1 Rules for stemming nouns 223
B.2 Structure of the general WordNet database with two microtheories: one extracted from *The Lord of the Rings* and the other extracted from *The Iliad.* Both depend on the basic WordNet taxonomy, which means that both have links to other synsets in it. 228
B.3 (a) In this taxonomy, *Aquila eliaca* is at the same time a concept and an instance [Welty and Ferucci, 1999]. (b) Although the student *John Smith* would normally be used as an instance, in some occasions it may be useful to consider it as a concept. 230
B.4 *Rama* should be an instance of *avatar*, but it is also a concept which has three different instances: the three incarnations. (Repeated from Figure 5.11). 233

C.1 Contents of the first page shown to a user that starts to browse the adaptive site about *The Voyages of the Beagle* in sequential order, for a profile on **history** (narrative text). 238
C.2 Contents of the first page shown to a user that starts to browse the adaptive site about *The Voyages of the Beagle* in sequential order, for a profile on **biology** (continued in next figure). 239
C.3 Contents of the first page shown to a user that starts to browse the adaptive site about *The Voyages of the Beagle* in sequential order, for a profile on **biology** (continued from the previous figure). 240
C.4 Same page shown for a profile on **geography** and descriptions (cont. in next figure). 241
C.5 Same page for a profile on **geography** (cont. from the prev. figure). 242
C.6 First section in the first chapter of *The Voyages of the Beagle*, summarised at 30% of its original size (continued in next figure). 243
C.7 First section in the first chapter of *The Voyages of the Beagle*, summarised at 30% of its original size (continued from the previous figure). 244

C.8 Page about the concept *Rio*, referring to Rio de Janeiro, in Brazil. Note that *Rio* is the Spanish and Portuguese for *river*, and even so all the information collected referred to the right meaning of *Rio* (16 more paragraphs remaining). 245
C.9 Page about the concept *Cornwall*. 246

D.1 Diseño de alto nivel de WELKIN. 257

List of Tables

1.1 Corpora used either for training or testing different modules of the system. 18
1.2 Sections that describe the main modules, and papers about them. 19

2.1 Ways in which different items from the knowledge base can be stored in a semantic network (see Figure 2.4) and as a FOPL theory. 36
2.2 Different texts generated depending on the user profile and the context [Not et al., 1998]. . . 40

3.1 Noun relationships in WordNet. Each relation is represented with a symbol. 52
3.2 Comparison of the sizes of some LKBs. The numbers are approximate, except in the case of WordNet 1.7. 55
3.3 Comparison of algorithms for creating or extending ontologies. The second column shows whether they are automatic or semi-automatic (when they require a human judge to validate their decisions or to name the generated concepts). The approaches marked as **semi* can be considered fully automatic, but due to the large amount of errors in the results, the authors recommend that a judge validates them. The third column shows the data on which they feed, and the fourth column shows the approach taken. 60
3.4 Comparison of algorithms for creating or extending ontologies. The second column shows whether they build an ontology from scratch or refine an existing one; the third column shows whether the learning is supervised, and the fourth column shows the methods used. 60
3.5 Comparison of different approaches to Ontology Refinement. The second column indicates whether the method is probabilistic or deterministic; the third column shows the ontology that was extended, and the fourth column is the corpus that was used to find new concepts. . 62

4.1 Schematic representation of a lexical matrix, from Resnik [1993]. 68
4.2 Example of lexical matrix, showing some words and the concepts they lexicalise. 68
4.3 Examples of near-synonym variation. Most are taken from Edmonds and Hirst [2002]. 70
4.4 Tokens that co-occur with the word *man* in the same sentence, and their frequencies of appearance. 72
4.5 Mutual information between verbs and objects; for the verb *to drink* [Hindle, 1990, from Resnik [1993]] 74
4.6 Values of the t-score when comparing words preceded by *strong* and *powerful*, from Church et al. [1991]. The second column shows the number of times that the word appeared after *strong*, and the third column shows the number of times that it appeared after *powerful*. . . . 74

4.7 Semantic similarity between five concepts taken from WordNet. The higher the number, the higher their similarity. 79
4.8 Similarity of the context of synset pairs, using the χ^2 estimate and the Kullback-Leibler distance. 79
4.9 Similarity of the context of synset pairs, using the χ^2 estimate and the Kullback-Leibler distance. 79
4.10 Similarity of the context of synset pairs, using the χ^2 estimate and the Kullback-Leibler distance. 79
4.11 Similarity of the context of synset pairs, using the χ^2 estimate and the Kullback-Leibler distance. 79

5.1 The concepts in the taxonomy, a hypothetical frequency for each concept, the results of adding up the frequencies of a concept's children, and the Information Content for every concept. . . . 89
5.2 Possible generalisations suggested by a classifier, and the values of the three metrics that take into account the Information Content of each node in the ontology. 90
5.3 Unknown synsets that appear in the sentence from Figure 5.7, and the WN hyperonyms proposed for each. 91
5.4 Corpora that have been annotated, so they can be used as test set for GNE Identification procedures. The columns list the size of the corpus (number of words), the criterion for choosing the terms (all unknown terms, or only those with appear a certain number of times), and the number of concepts that were selected for each corpus. 91
5.5 Some of the unknown nouns identified in the domain-specific document, ordered by frequency. The third column specifies which of them are concepts and which are instances. Finally, the last column displays the synset to which the unknown synset should be attached. It is the purpose of my framework to produce that attachment. 97
5.6 Some of the unknown nouns identified in *The Iliad*. 97
5.7 Some top words in the signature of the Dwarf. The second column is the frequency count, and the third column is the weight of the word, using Yarowsky's function (w_1) and Agirre's function (w_2). 102
5.8 Topic, subject, object and modifier signatures of the concept *person*. These are the top frequency words, together with their weights. A higher frequency does not imply that the weight will be higher, because the word may be equally frequent for every synset. Words with weight zero are more frequent in other concept's signatures. 103
5.9 Similarity values for each of the decisions that have been taking when classifying the unknown concept *hobbit*. In the first place, when deciding between *entity* and its children, the winner synset was *being, life form*. In the second decision, when deciding between this last synset and its children, the winner was *human*. Both decisions were correct, because they were the synsets more similar to the meaning of *hobbit*. 105
5.10 Comparison of two methods to combine the results provided by the signatures. Columns represent: strict and lenient accuracy; Learning Accuracy; the percentage of times that the algorithm chose the correct decision (C.D); and the mean position of the correct decision to choose (M.P.) 106
5.11 Results using different signatures. 106
5.12 Comparison of my approach with two different systems. The lines labelled TH and CB show two improvements performed to the basic algorithm by Hahn and Schnattinger [1998] 107

6.1 Results without and with patterns. The columns represent strict and lenient accuracy; Learning Accuracy; the percentage of times that the algorithm chose the correct decision (C.D); and the mean position of the correct decision to choose (M.P.) 116
6.2 Similarity values for each of the decisions that have been taken when classifying the unknown concept *Frodo*, and factors provided by the hyponymy patterns. 117
6.3 Results (precision and recall) for the identification of events in the sample texts. 120
6.4 Temporal relation between *going* and other events in the example sentences in (13). 121
6.5 Results for finding temporal expressions in the documents 122
6.6 Results for finding temporal relations between events, using the verb tenses and the temporal expressions. 123
6.7 Typical endings for the genus and the species of a scientific name. 124
6.8 Recall and precision values for classifying pairs of words as Latin or English. The pairs of words were 200 scientific names, and the first 200 words from *The Voyages of the Beagle*. . . 125

7.1 Four rhetorical relations from Mann and Thompson [1988], taken from Mani [2001]. The text in *italics* contains the nucleus of the relation. 139
7.2 Example of macro-rules (from Mani [2001]). 143
7.3 Types of relationships across documents (from Mani [2001], from Radev [2000]). 145

8.1 Top-frequency words in the signatures for the narrative and descriptive texts. 152
8.2 Top-frequency words in the signatures for the biological texts. 152
8.3 Results classifying the paragraphs in one of the three topics: biology, descriptive and narrative. 154
8.4 Evolution of the population of summaries. 163
8.5 Agreement between pairs of judges. 164
8.6 Level of agreement between judges. 165
8.7 Agreement between each judge and the automatically generated summaries. The last line compares the summary formed with the sentences that received more votes from the judges against the automatic summary. 165
8.8 Adaptive navigation support techniques that have been implemented for the user interface of the system. 168
8.9 Adaptive presentation techniques that have been implemented for the user interface of the system. 170
8.10 Results for the multi-document summariser for information collected from the Internet. Between parenthesis are the number of selected pages and paragraphs that were incorrect. . . . 177
8.11 Reasons for the recall errors committed by the algorithm. 177

9.1 People that appear in the text, and the way in which they were classified. 187
9.2 Performance of each of the modules that are executed in order to create automatically a hypermedia web site from a linear document. 188
9.3 Performance of each of the modules that are executed in order to create automatically a hypermedia web site from Osler's *The Evolution of Modern Medicine*. 195
9.4 Words in the topic signatures, taken from the first 100 paragraphs in Darwin's *The Voyages of the Beagle*, and words taken from the first 38 paragraphs from Osler's *The Evolution of Modern Medicine*. 195

9.5 Performance of each of the modules that are executed in order to create automatically a hypermedia web site from Hegel's *Lectures on the History of Philosophy*. 198
9.6 Profiles of the users. Each question was evaluated from 1 to 5 (very low, low, medium, high and very high). 200
9.7 Answers of the users. Each question was evaluated from 1 to 5 (very low, low, medium, high and very high). 201
9.8 Profiles of the users. Each question was evaluated from 1 to 5 (very low, low, medium, high and very high). 202

B.1 Some part-of-speech labels. 223
B.2 The first rule in the Transformation List for labelling base Quantifier Phrases. 224
B.3 The first rule in the Transformation List for labelling base Noun Phrases. 225
B.4 Results of the manual annotation of instances and concepts in WordNet (only the 51,553 leaves are considered; the rest of the synsets are all considered concepts). 232
B.5 Results of the five-fold evaluation 235

D.1 Corpora utilizados para entrenar o probar los distintos módulos del sistema. 261
D.2 Secciones que describen los módulos principales, y artículos sobre ellos. 262

Abstract

It is a fact that the Internet has consolidated as a widely used mean to convey information. It was soon appreciated that different people access the web with different needs, a fact which motivated the appearance of web sites that provided different information and were structured in different ways depending on the user. Nowadays many web-based systems store user profiles containing some characteristics of the users. These profiles are used to decide which particular information will be shown to each particular visitor, and how it will be organised.

Moreover, different kinds of applications need to know different characteristics of the users. For instance, e-commerce applications use the shopping history and the user's tastes in order to suggest further products; on-line educational systems keep track of the concepts that have already been studied, and the tests that have been successfully solved by the student; and on-line information systems and retrieval applications have to know precisely the information needs of the user in order to provide the most relevant data. In the same way, the procedures for deciding the contents and structure of the web sites in function of the user profiles vary across applications.

Even though there are applications for authoring web sites, constructing them is not yet particularly easy. Amongst the limitations of current authoring tools for on-line information systems are that the kinds of information stored in the user profiles or the rules for adaptation are usually restricted to a few pre-defined types; but, most importantly, they usually require the web author to write all the particular chunks of texts that will be presented to the different users. Therefore, the web author probably has to write as many different versions of the same texts as the number of possible user profiles that affect the contents of the site.

This work describes a framework that combines techniques from different fields in order to create, in a fully automatic way, on-line information systems from linear texts in electronic format, such as textbooks. It borrows ideas from User Modelling and Adaptive Hypermedia for storing and updating the user profiles, and for changing the contents and the structure of the web site according to them. Natural Language Techniques are also applied in order to gather automatically information about the relevant terms found in the original texts, and for adapting the output contents of the site, using automatic filtering and summarisation techniques. The architecture is divided into two steps: an *off-line* processing step, which collects information about the original linear text, and an *on-line* step, which executes when a user connects to the system with a web browser, and the contents and hyperlinks are generated.

The framework has been implemented as the WELKIN system, which has been used to build three adaptive on-line information sites in a quick and easy way. Some controlled experiments have been performed with real users aimed to provide positive feedback on the implementation of the system.

Resumen

Internet se ha consolidado como una herramienta muy útil para transmitir información. No tardó en observarse que distintas personas la utilizan para satisfacer diferentes necesidades, lo que provocó la aparición de técnicas para, según el usuario, proporcionar información diferente y estructurar las páginas de manera distinta. Hoy día se pueden encontrar muchos sistemas basados en Internet que almacenan internamente las características de los usuarios en perfiles personalizados. Cuando los usuarios se conectan al servidor, estos perfiles servirán para decidir qué información seleccionar para cada uno, y cómo organizarla.

Además, los distintos tipos de aplicaciones necesitan recordar datos diferentes sobre los usuarios. Por ejemplo, las aplicaciones para comercio electrónico recordarán el historial de compra y los gustos personales para poder sugerir otros productos; los sistemas de educación por red recuerdan qué conceptos ha aprendido el estudiante y qué problemas ha resuelto; y los sistemas de información en línea y los sistemas de recuperación de información han de saber las necesidades de información del usuario para proporcionar los datos más relevantes. Del mismo modo, los procedimientos para decidir cómo seleccionar los contenidos y la estructura del hipertexto en función de los perfiles de usuario difieren según la aplicación.

Existen herramientas de autor para ayudar en la construcción de aplicaciones de hipermedia adaptativas, pero aun así suele requerir bastante trabajo crear una. Algunas de las limitaciones de las herramientas existentes son que las características que pueden modelizarse de los usuarios y las reglas de adaptación suelen estar bastante restringidas; además, en general requieren que el autor escriba todas las porciones de texto con las cuales se crearán las páginas generadas. Esto implica que es necesario escribir tantas variaciones del mismo texto como perfiles de usuario que puedan afectar al estilo del texto.

Este trabajo describe una arquitectura que combina técnicas de distintos campos para crear, de manera completamente automática, sistemas de información en línea a partir de textos lineales en formato electrónico, como los libros de texto. La investigación toma ideas de los campos del Modelado de Usuario e Hipermedia Adaptativa, para la creación y utilización de los perfiles de usuario. También se utilizan técnicas de Procesamiento de Lenguaje Natural para recoger automáticamente información sobre los términos relevantes encontrados en los textos originales, y para adaptar los contenidos del sitio web a los diferentes usuarios, mediante filtros y resúmenes automáticos. La arquitectura se divide en dos pasos: uno realizado *fuera de línea*, durante el cual se recoge información sobre el texto lineal original, y otro realizado *en línea*, que se ejecuta cuando cualquier usuario se conecta al sistema con un navegador, momento en que se generan los contenidos de las páginas y los enlaces entre ellas.

Este entorno se ha implementado en el sistema WELKIN, que se ha utilizado para construir varios sitios de información en línea de manera sencilla y rápida. Se han realizado algunos experimentos controlados con usuarios reales para recoger sus opiniones acerca del sistema.

Acknowledgements

First of all, I would like to thank my parents and my sister Mariangel for all the support and help they have given me. This thesis is dedicated to them, without whom this work wouldn't have been possible.

I would like to express my gratitude specially to Pilar Rodríguez, for her time, patience and work, and for her advice when taking important decisions. It has been a pleasure to work with her, and I hope to continue doing so in the future.

Also my thanks to Roberto Moriyón, the tutor of my first scholarships and a constant help whenever needed, and to Juana Calle, the secretary of the department, for her kindness and efficiency.

My special thanks as well to Suresh Manandhar, from the Computer Science Department at York, who gave me my first lectures in Natural Language Processing and not only supervised my research stays in England, but also helped with the usual problems of being in a foreign country.

I would also like to thank all the people which have accompanied me in the last four years in my life, and I would like to mention specially Alfredo Escalona, who always gave me his friendship, both in York and in Madrid; and to Amaya, César, Pedro, Lara, Belén, Jorge and Marino; to Germán Montoro, Miguel Ángel Mora, Rosa Carro, Álvaro Ortigosa, Juan de Lara, Alfonso Ortega, Ruth Cobos, Pablo Haya, Pedro Paredes, Leila Shafti, Abdel Latif, and Manuel Freire from the *309, B-406 and B-207 laboratories*; to Jordi Porta, for the long talks held in the cafeteria of the department; to Cristina, Laura, Christian, Clara, Celia, Ana María, James, Beatriz, Dani, Ana and Raquel for all the good moments lived in the *H house* at Saint Lawrence Court, and to all the other housemates I've had all this time; to Fernando and Josefina, José Luis and Karin, Eduardo and Esther, Luis Murillo, Maria Cubel, Jordi Coral, Roberto León, and Fr. Tony from York; to Imma, Marc, Diana, José Antonio, Estefanía and Estrella who helped in the evaluation of the summaries, or used WELKIN and provided me with very good ideas for improvement; to all the people in the *Ghia* group, and all the current and former members in the deparments of Computer Science both at Madrid and York.

Finally, my very special thanks to María, for all the shared moments in these years. This thesis is not mine any more but ours.

Part I

Introduction

Chapter 1

Introduction

The World Wide Web has made hypermedia a widely used mean for conveying information. However, the quantity of data available on the Internet grows steadily, and the large amounts of information can pose potential problems to the web surfer [Wu and de Bra, 2002]: firstly, static web sites offer the same information to all kinds of users, independently of their interests (the so-called *"one size fits all"* paradigm). In this situation, it can be difficult for users to find relevant data, or they may be forced to spend some time browsing uninteresting information before finding it. Secondly, web pages can also produce *comprehension problems* to the user, because the author of the pages is making implicit assumptions about the user's previous knowledge. This may provoke that users find that some piece of information is too basic for them, if they already knew it before, or that visitors cannot understand the contents of a web page because they do not have the required background.

This fact has motivated research in many different areas. On the one hand, there appeared *Adaptive Hypermedia* (AH) applications that try to provide the users with the information they need, and to help them in finding their way from document to document. A few examples are Information Retrieval applications [Baeza-Yates and Ribeiro-Neto, 1999], that search for information inside huge repositories of documents or in the Internet, according to some user's profile or query; adaptive hypermedia educational systems, that help the students to navigate across a course in function of the concepts that they have already learnt; or on-line information systems, that try to show the users the information that they need according to their interests and context (e.g. museum information systems).

On the other hand, there is a wholly different line of applications, thought to ameliorate the problem of the information overload, which stemmed from research on *Natural Language Processing* (NLP). Here, we may cite *Information Extraction* applications, that obtain structured information from textual documents; *Question Answering* systems, that look for the answer of a question written by a user in a collection of documents, or *Text Summarisation* applications, that condensate the relevant information found in a larger textual source. In most cases, *Information Retrieval* (IR) applications use tools from the field of computational linguistics, such as stemmers, which obtain the root of inflected words. Therefore, we can also include IR in this group of applications.

The semantic web initiative [Berners-Lee et al., 2001] is working on standardising web languages such as RDF or DAML+OIL so the hypertext pages include semantic information about their contents. In this framework, web pages would include semantic information indicating whether they refer, for instance, to a person, a book, a film or a restaurant. Until these markup languages and web services have been defined

and accepted, automatic techniques can be used to discriminate between the retrieved web pages. A word sense disambiguator is a system that identifies the meaning with which a word is used, in some particular context [Ide and Véronis, 1998]. If the sense of the words in the user's query, and the sense of the words in a document are disambiguated, then it should be possible to discover in which documents the query words are used in the sense intended by the user.

Information surfeit is a real problem, and many potential solutions are being explored looking for ways to handle it. Not only on the Internet there are large quantities of data; the problem also arises, to a smaller scale, in personal computers. The current capacity of hard disks allows users to store volumes of information that would have been impossible on hard copy, and a search to find some information even in "small" hard disks may take several minutes to complete. Many different possible applications can be attempted in order to help the user find relevant information. The applications described above, using AH and NLP techniques, are only some of the many different systems that can be deployed. Below there are listed a few examples, some of which are already prototypes or commercial systems, while others can be expected further ahead in the future:

- Extracting and structuring relevant data from documents which have information about a specific domain or follow a similar structure, such as finding information in newswire articles or matching curricula vitae with job advertisements.

- Filtering irrelevant information and highlighting relevant data, e.g. removing spam and advertisement e-mail messages from the mailbox, or choosing documents that look important for a user from a distribution list.

- Extracting and structuring relevant information (e.g. phone numbers, addresses, or important reports) from heterogeneous data, such as e-mails in mailboxes, so it is easier for the user to retrieve it later.

- Planning systems that decide and perform all the steps involved in fulfilling a user's need. For example, if a user wants to go to some conference, a planning system might obtain directly from the Internet information about the date of the conference, flights and hotels available, and generate one or several travel plans; and the user's agenda might be checked and rearranged if there were meetings or other tasks scheduled for those dates.

- Question-answering systems that respond to user's questions and petitions, although the process may involve some kind of semantic reasoning with a Knowledge Base, such as making common-sense deductions or showing a theorem.

This work centres on automatically building hypermedia sites that are adapted to the needs of the particular users. The following sections further specify the task addressed: firstly, Section 1.1 describes the motivations for attempting the work. Next, Section 1.2 describes the requisites with which it must comply. Sections 1.3 and 1.4 describe the design of the theoretical framework, and the different modules of the practical application that has been deployed according to the guidelines of the framework. Section 1.5 describes the different textual resources that were used as training and test data for each of the modules, and the texts which have been used as source material to construct some example adaptive sites. Finally, Sections 1.6 and 1.7 contain, respectively, a summary of the contributions of this work, and an overview of the thesis.

1.1 Motivations

As already said, AH appeared as a mean to overcome the *one size fits all* paradigm, providing the framework needed to tailor web pages to the needs of the specific users. The adaptation is usually based on a *user model* that encodes some of the user's characteristics, such as preferences and previous knowledge, and a *device model*, which stores the characteristics of the device used, such as screen size or network connection. These models are used in order to provide personalised information. Popular techniques for AH include *adaptive navigation support*, consisting in adapting the link structure of the web site so that the user is guided toward interesting information; and *adaptive presentation*, consisting in adapting the contents of the web pages to the user's needs, e.g. by hiding irrelevant paragraphs and highlighting the fragments that the system believes will be more relevant [de Bra et al., 1999a].

These techniques, while simplifying the labour of a web visitor, implies, on the other hand, a large increase in the amount of work for the hypermedia author. Now, it is not enough to write the contents of a web site and to connect the different pages together with links, but it is also necessary to define the characteristics of the user that are to be modelled, and the rules that will determine how to present the contents to different users and how to guide them through the site. One of the answers to this need has been the appearance of authoring tools for adaptive web sites [Brusilovsly et al., 1998, Murray et al., 2000, Sanrach and Grandbastien, 2000]. They reduce the design work by providing a framework in which several characteristics are more or less fixed, such as the user profiles and the adaptation mechanism, but still the designer has to work much in order to explore and take advantage of all the possibilities for adaptation.

1.1.1 Problem to solve

One of the problems of the large amounts of information available is that there is hardly ever enough time to read all the documentation about some specific topic. Documents are sometimes large and multi-disciplinary, and include paragraphs and sections about different matters. A user with little time available who needs to gain a background on a certain specific area, can possibly find that the information is dispersed in different books, and scattered in different places in the same books. For this particular situation, it would be useful to have an automatic procedure to select information from separate sources, put it all together, according to some user's profile, and provide an internal structure with separate sections and hyperlinks between these sections.

For example, let us imagine that a man has to learn about the interface between two different software tools, such as the programming language Java and the relational database management system `mysql`. If all the work were done manually, the procedure would be the following: firstly, he would identify some textbooks about the topics, such as a programming manual in Java, a `mysql` user's guide, and a manual provided with a JDBC driver for `mysql`. Then, with a look at the indexes of the documents, or with a quick browse, he would analyse the data, compare it with his interests, and mark the most relevant sections for his purposes, such as the portions of the Java manual that refer to relational databases management (containing the description of the `java.sql` package); the portions of the `mysql` manual that refer to interfacing the database with programming languages; and the JDBC drivers' manual. When this is done, he will peruse the relevant sections. Maybe, he will complete them with some additional information, possibly found in the Internet, or in documents about JDBC Frequently Asked Questions and discussion lists of people who had problems whilst using the software.

The work proposed in this thesis is intended to mimic the human's actions when looking for relevant information that satisfies some particular need:

1. Information can be collected from domain-specific sources, such as books and articles about the topic in which the user is interested, and which are provided by the user. Expectedly, there will be large amounts of relevant information in these sources, as they are carefully selected by the user as relevant source materials.

2. The selected data can also be extended with information taken automatically from open general-purpose corpora, such as the Internet.

3. This information can be shown to the user in a structured way, with tools that help to reduce the information selected while keeping the most relevant pieces, and which help in navigating through that information.

In order to build such a tool, it has been necessary to investigate the following:

- Ways in which information about the user's interest can be represented and acquired.
- Ways in which information can be selected, from different kinds of sources, according to the user's needs.
- Ways in which the information, once tailored for some user's need, can be structured and organised so the user can access it.

1.1.2 Linguistic Motivation

In order to create the adaptive site from a set of source documents, it is necessary to analyse, to some extent, the information provided in the texts. Although some tasks, such as those that can be expressed in terms of mathematical computations, traditionally have been relatively simpler to automatise with computers, others, specially the ones that require a large amount of knowledge, still have to be performed by humans. It is said that they suffer from the knowledge acquisition bottleneck [Hayes-Roth et al., 1983], because the formalisation and encoding of knowledge in a machine-understandable way is a hard task. The problems that involve a processing of unrestricted texts is a typical example of these difficult tasks, because it is usually necessary to encode some amount of world knowledge and, in general, they only perform well in restricted domains.

On the other hand, in the last years, several commercial applications have been launched that perform some preliminary textual analysis, such as retrieval applications for search engines or Text Summarisation systems for text editors. Although these still have a large number of errors, their accuracy increases as research progresses.

In the understanding of natural language there are some problems that have produced or still produce a bottleneck in the research. One of these hard problems was syntax analysis or parsing, which has received much attention in the last years. Now, there are practical applications with reasonably good results, at least for the English language. Apart from hand-coded syntax analysers, the availability of collections of parsed documents, called treebanks, means that accurate parsers can now be automatically constructed with machine learning procedures [Hockenmaier and Steedman, 2002, Xia, 1999]. However, there is still work to do, for instance, in adding robustness to parsing ungrammatical sentences.

Concerning the semantic analysis, discovering the meaning of sentences is a task that is receiving much attention lately, with the help of robust parsing and conceptual semantic networks such as WordNet [Miller, 1995]. Research is also addressing already the fields of dialogue interfaces and pragmatics, which need some degree of accuracy in the semantic analysis in order to provide practical applications.

Many of the applications of linguistic processing can be used in adaptive hypermedia. These include dialogue interfaces, question-answering systems (systems that look for users' questions in a set of documents), text summarisation, natural language generation and many others. This is the reason why research on computational linguistics is also an important motivation for this work.

1.1.3 Goals and expected results

The large amounts of information available on the Internet, and the lack of coherent indexes, force the users to spend much time looking for the information they need. Several studies conclude that employees spend much of their time looking for information. Just as a couple of examples, McKinley [1997] states that 20 percent of administrative time may be spent filing and retrieving important documents; and, according to a MORI Research Poll released by Mediapps on 05/09/2000, employees waste thousands of hours of their companies' time searching for relevant business information on the Internet, costing companies tens of thousands of pounds a year[1]. Considering this fact, an adaptive system that helps a user find information on some topic, either in a small collection of specific documents or in the Internet, can be very useful, as it reduces the time needed to find the information, and the cost it means to the companies.

Although that is not always the case, the design of adaptive hypermedia sites can sometimes be divided into two steps. Firstly, there is an **off-line** step in which all the information that will appear in the hypermedia site is collected and structured. This may involve collecting all the textual information that will be present in the site, and generating a Knowledge Base about the concepts described in those texts, that can be used, for example, to track which of the concepts are already known by the user and which are not, and to make inferences about whether some kind of information can be shown to the user in some particular moment. This off-line tasks include the following:

- *Identification of the relevant topics and the future sections of the site.* If the author is already an expert, then this step might not pose much complication; otherwise, it may involve a strong interaction with domain experts. This task is common with non-adaptive hypermedia design.

- *Writing the contents of those sections.* The author has to write textual units describing the several topics. In contrast with traditional web construction, it may be necessary to write different versions of the same topics, such as equivalent texts in different languages, with different lengths, or intended for people with different cultural background.

- *Generating the Knowledge Base,* if it is necessary for the application, indicating which are the relationships that hold between the textual units. For example, some text might contain a description of something more specific than other text, and in that case it may be necessary to state that the second text has to be already visited before allowing the user to read the first one.

Secondly, there is an **on-line** step which functions when the users access the adaptive site to find information. For this step, it is necessary to perform the following work:

[1] http://web01.mediapps.com/web/uk.nsf/$$pagesweb/CompanyPressReleases2

- *Adapting the contents* according to the user's profile or environment. This may include showing different fragments of texts (e.g. written in different languages or with different levels of detail), changing the media (e.g. providing either a text or a image containing the same information), and several other techniques.

- *Adapting the structure of the site* in order to guide the user in the search for information. This includes hiding or showing links if they are considered irrelevant or interesting, respectively; creating new links on-the-fly; annotating the links so the user knows the kind of information to which they lead; etc.

> The final goal of this thesis is a complete automation of all these steps, so that the job of the designer is reduced to a supervision of the system. Using a set of texts that is supposed to contain the contents of the future hypermedia site, the system has to identify the relevant sections, to select the text included in each section, to structure the different pages as a hypergraph, and to provide adaptation mechanisms for different users.

There are two different products of this work. Firstly, the description of a theoretical framework that combines ideas from different areas in order to fulfil the goal of automatically generating the web sites. Secondly, the implementation of the architecture as a system which has been called WELKIN. In the future, we shall use the name WELKIN to refer to the particular implementation of the framework. Appendix C shows some example pages that were generated automatically for different users. The following sections describe these tasks in more detail, and the hypotheses that have been taken for implementation purposes, in order to simplify them.

1.2 General purposes of the research

This section describes the requisites that have been defined for this work, in order to address the general problem stated in the previous section. At the present state of the technology, if the framework were too general it might result unfeasible, so it is necessary to restrict the features and the reach of the final architecture.

Operational requisites

The general objective of this work is the identification and presentation of relevant information to the users, addressing the need that stems from the information overload. The relevant data should be structured as a hypermedia site.

The input material consists of the following elements:

1. One or several texts, which are chosen by the users, about a domain which is relevant to their interests. For example, a user interested in zoology and the origins of Darwin's theory of evolution by natural selection might select Darwin's books as source data.
2. A connection to the Internet, from which additional information will be collected, if necessary.
3. A description of the user's interests and goals.

The output material is a complete adaptive hypermedia web site, constructed from information found in the original documents, and in the Internet. The site should only show to each user the particular information

that is considered relevant according to the internal description of the user's goals, and internally structured in sections and hyperlinks amongst them. The hyperstructure should help the user to navigate in search for useful data.

The operational requisites that this generated web site should satisfy are the following:

- It should provide simple ways in which the users can describe their interests.
- It should supply relevant information.
- This information should be structured as a complete web site, with appropriate links to navigate the information.
- It should also provide means to adapt the information to different users, or even to the same user with different requirements, such as a higher or a lower level of compression to be performed on the information.

Purposes of adaptation Multi-disciplinary knowledge sources contain information relevant to different fields of knowledge. It is often the case that a user has to access a multi-disciplinary text while being interested only in some part of it. The adaptation of contents will be a very important feature of the architecture. The input information should be annotated with the necessary information so it can be easily selected for presentation for a user interested in it.

The adaptation purposes can be summarised in the following points:

- It should help users to discriminate between relevant and irrelevant information, by creating a user model that reflects their interests, and highlighting the text sections that are deemed more relevant.
- It should tailor the text presentation to users' needs, by adapting the quantity of information to the time they have available.

User Model descriptions The architecture has to encode information about interests and goals in the user profile. In some fields, such as Information Retrieval or Question Answering, the only data that the application knows about a user is a certain query. When the user writes a second query, the first one is lost, so the user's profile changes completely each time the system is used. On the other hand, on-line information systems usually store more static characteristics of the user, such as knowledge and interests.

In this particular case, the user is interested in finding relatively large quantities of useful information inside a much larger repository of data. The output of the system will be a web site that might take some time to read. Therefore, the user model has to store more stable characteristics of the users, such as their interests.

On the other hand, the generated web site is intended to satisfy some temporal user need, and it is likely that, when the whole site has been read and the need is fulfilled, the user's profile is not necessary any more. In other words, users' characteristics such as previous knowledge, the pages that have already been visited, or results of evaluations to check that they have learnt the material are given less attention, as it is not necessary to model them. Once the on-line information site has been generated, the users are free to browse whatever they like, and there will not be a control on whether they really read and learn the information, as there is in Educational Systems.

Input for user model acquisition The main function of the user model will be to store user's interests for different themes that may appear in the documents. There are several possible ways in which the user

model of interests can be acquired:

- With a set of pre-defined general topics that can be found in the original documents, such as the main divisions of sciences and arts. In this case, a user might be described with a predefined interest (a stereotype) or a combination of them. For example, during registration, a user might declare interest on history and computer science.

- By asking the user to provide a set of documents or paragraphs which are relevant, or to classify in order of relevancy some documents provided by the system. This allows the user to have a personalised user profile from the very beginning.

In any case, it is desirable that the user or the system be able to modify dynamically the interests profile depending on the actions performed, the pages visited and the items for which interest has been shown.

Adaptive Hypermedia models

Traditional adaptive hypermedia applications can be divided into three sub-models [Wu and de Bra, 2002]: the *domain model*, which contains information about the contents of the site (such as fragments of texts or images) and their structure (i.e. the rules that define the relationships between the sections); the *user model*, that contains the information about the user (this includes the device used); and the *adaptation model*, that contains the rules which decide how to display the contents, and which contents are to be shown to each particular user.

Inside this framework, the main purpose of the proposed architecture is the automation of the domain model: the contents of the hypermedia site, and the rules which state which contents can be presented and in which order. This has also implications on the *user model* —because the contents are dependent on the user's interests and goals— and on the *adaptation model*, because the design of the site depends to some extent on the ways in which it has to be adaptive.

Areas of application

The generation of any possible kind of text from any domain is currently out of the scope of this work. Different fields of knowledge use different terminologies; the documents follow different layouts; they may include tables, figures, examples, programming code or pseudo-code, diagrams and many kinds of information each of which has to be processed with different tools.

On the other hand, it is possible to define a general architecture. In such a framework, different modules, able to process different kind of information, could be *plugged in*, and their output could be used by the hypermedia site generator in order to decide whether each piece of information is useful or not. For example, in computer science, the relevant terminology that has to be identified consists almost entirely of technical names; while in biology it consists of animals and plants, scientific names, proper names and even locations and dates; and, in philosophy, it may consist mainly of abstract names. Therefore, the modules that recognise unknown terms may have to be substituted if the system is ported to a different domain. In the same way, when processing a computer science text, it might be useful to distinguish the explanations, written in natural language, from the example codes. The following ideas describe a way to meet this requisite:

- The architecture has to be modular, and different modules should be interchangeable.

- There should be a way in which different components can provide information to other modules inside the global architecture, such as adding annotations in the source documents.
- There also should be a way for indicating whether some module requires the annotations produced by other particular module. For instance, a module that identifies relevant terminology might require that the noun phrases have been identified previously in the texts.
- For different kinds of texts, it should be possible to define alternative modules that perform the same function, but specialised on different domains, such as a Term Recogniser for biological and computer science terminology, or for processing texts in different languages.

Interface requisites

The proposed output is a collection of sections structured as a hypermedia site. Considering that the most popular hypermedia environment nowadays is the World Wide Web, this output should be produced in HTML format, which can be viewed with the standard web browsers, from any machine that is connected to the Internet. As in any hypermedia application, the information is structured as hyperdocuments and hyperlinks between them. Furthermore, some adaptive hypermedia methods and techniques should be used as well, such as the following:

- **Adaptive contents**: *adding or removing text fragments* according to the user's interests; *reducing or expanding* the information in function of the user's availability of time, and using Natural Language Processing techniques in order to produce *summaries* of the texts adapted to the user's profile.
- **Adapting navigation support**: *link adaptation* for creating links on-the-fly according to the user's preferences; or *link annotation* by colouring the links according to the type of the information toward which they lead.

1.3 Architecture: theoretical approach

This section describes a possible way in which the requisites described in the previous section can be fulfilled in a single framework. The processing has been divided into two parts, as indicated in Section 1.1. The first one, executed off-line, collects all the material necessary to find the texts that are important for each of the relevant topics of interest. The on-line step includes all the actions that must be taken when users enter the site looking for information, in order to provide the data that is more appropriate to each of them.

1.3.1 Contents identification (off-line step)

The first step in designing an adaptive hypermedia site is the decision about the kind of information that it will contain. There are, often, linear texts, such as books, manuals or articles about that topic that can be used for obtaining information for the site. *The first hypothesis* that is taken is that such kind of textual sources will be available about the web site designed. Next, it will be necessary to decide how the information from those linear texts will be arranged as a hypermedia site, i.e. how the information from the texts will be divided into separate hyperdocuments in order to create the site.

Secondly, in every particular domain there is a set of words that are not used in common language (and which therefore do not appear in dictionaries), which describe domain-specific concepts. For example, in

a text about history this restricted terminology includes the names of the characters, locations, political parties and tendencies, countries, etc.; in a text about mathematics, it includes the names of theories, theorems and algebraic entities, amongst others; in a text about biology, it includes taxon and location names, biochemical compounds, etc. *The second hypothesis* taken is the following: a complete adaptive web site can be constructed with the information from linear text (e.g. a textbook) where all hypermedia documents are of the following three types:

- The sections already present in the source texts.
- Hypermedia pages containing information about commonly-used domain-specific terms.
- Index pages that list these term-specific pages in an ordered way.

For example, if we have a written report enumerating the research performed in a University department, which is structured according to some administrative guidelines, the following hyperdocuments could be extracted from it:

- Web pages summarising each of the sections in the report: introduction, different kinds of research performed, funding, conclusions, etc.
- Different pages for each of the specific terms, such as scientific terms referring to fields of study, personal names of the researchers involved, locations where research stays have been performed, etc.
- Index pages, that include the index of the original report, and lists of faculty members and scientific areas.

Of course, this hypothesis is restricting the kind of hypermedia pages that can be automatically generated. For instance, it does not allow for hypermedia pages such as site maps, or pages that describe more than one specific term at the same time. However, it will be assumed that the proposed three kinds of pages are sufficiently versatile for building useful hypermedia sites.

Using the second hypothesis, an important processing that has to be done is the identification of domain-specific terms, a task that is called Term Extraction (TE) [Cabré et al., 2001]. Sections are usually clearly marked in linear texts, and the index pages are enumerations of the terms that have been identified. TE is a hard task that has received much attention and is not completely solved yet. The usual methods for identifying domain-dependent words usually rely on the assumption that domain-dependent terms appear relatively more frequently in the texts from that particular domain than in a general text collection. However, that is not always guaranteed.

After identifying the relevant terms, it will be useful to find (or, at least, to bound) the meaning they convey. There are many different formalisms in which semantics can be encoded in a lexicon. A few examples are logic formalism, such as Description Logics [Borgida, 1996]; Knowledge Representation Systems, that make automatically inferences about the knowledge encoded, such as Classic and NeoClassic [Patel-Schneider et al., 1996], or LOOM [MacGregor, 1990]; or conceptual semantic networks. These are graph structure in which nodes represent concepts, and arcs represent relationships between them, such as IS_A, IS_A_PART_OF, or IS_THE_PURPOSE_OF. It is possible to make inferences about the meaning of a concept if its position in the semantic network is known.

The output of this task will be an enumeration of the sections that the web site will contain, and some information about their semantics, such as the kind of entity to which each section refers (e.g. a person, a location, an artifact, etc.)

1.3.2 Contents generation (on-line step)

Given that hypermedia consists of documents and links between them, there are two kinds of methods that can be used to adapt a web site to different users and devices, as stated before: adaptive presentation consists in showing different contents such as providing the same textual information in different ways or changing the textual information depending on the user interests; and adaptive navigation support consists in changing the link structure to help the users in finding useful information.

Concerning adaptive presentation, once the relevant sections have been identified, the next task consists in providing the contents to each of the sections. Index pages can be lists of links to sections that are related in some way, such as a list of terms that are theorems, or a list of terms that refer to people. Secondly, the generation of the section bodies can be done in many ways, such as selecting the fragments from the domain-specific documents that contain information about that section, or combining information obtained from a diversity of sources. The original fragments taken from the linear text can be processed so as to select the information that most likely will be relevant for each user. This can be done in many ways: by hiding fragments of information, by highlighting portions of the generated page, or by automatically generating a summary of the original text tailored to the user's interests.

In general, all the procedures for summarisation can be classified in two groups: extracting and abstracting [Mani, 2001].

- **Extracting** consists in literally copying material from the source documents. A typical procedure divides the text in portions (e.g. sentences or paragraphs), and evaluates each portion with a relevance metric. The ones that receive the higher score are extracted and put together in the *extract*.
- **Abstracting**, on the other hand, is performed when the summary includes some material that was not present in the source documents. It will probably require some paraphrasing of the sentences, and the resulting information is more condensed. An *abstract* may also be generated as a second step after producing a *extract*, by rearranging the information that had been selected.

In this work, in order to generate the contents of the sections, two different sources will be used: the original domain-specific texts, from which the core of the information will be taken, and the Internet, in order to extend the original information with extra data for the users who are interested in it.

The adaptive navigation support will include all the techniques used in order to link these content pages to each other so as to facilitate the user find the information needed.

1.4 The WELKIN implementation: high level design

This section includes a high-level description of the implementation, which has been called WELKIN[2], and a brief description of its components. In the remainder of the thesis all the modules will be described, with details about their crafting and the evaluation procedures.

The internal structure of WELKIN is represented in Figure 1.1. The input consists of three resources: one or several documents, a conceptual semantic network, and the Internet (represented at the right-hand side of the Figure). The processing can be divided into two steps: one that is performed off-line, by examining the original texts, annotating them and acquiring information from them; and the second one, performed on-line,

[2] *Cloud, sky*, in Middle English, from the Old English *wolcen*. Nowadays used as *sky* (*"ring the welkin"*). The name of the architecture refers to the fog of information inside which the relevant data is hidden and has to be found.

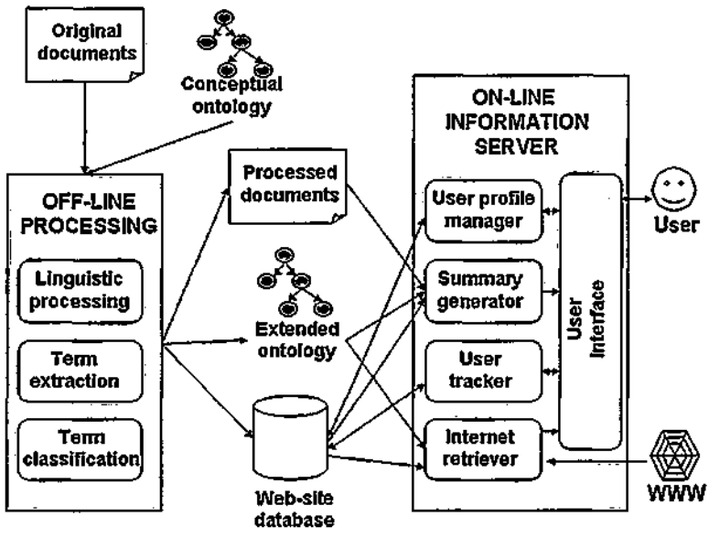


Figure 1.1: Overview of the architecture for generating content pages for an adaptive site.

which happens when a particular user enters the on-line information system looking for a hypermedia site based on those texts, and adapted to his or her interests.

1.4.1 Off-line processing

During the **off-line** processing, the domain-specific texts provided by the user are analysed with some linguistic tools, and then examined in order to extract the relevant terminology from them. This terminology will be used later, when the sections of the web site are planned and structured. WELKIN processes the documents with the following modules:

Linguistic processing

The texts provided by the user are processed with several tools for linguistic processing, described in detail in the Appendix Section B.1. They include the following:

- A tokeniser that identifies word boundaries.
- A sentence splitter that finds sentence boundaries.
- A stemmer, that returns, for each inflected noun or verb, its stem.
- A part-of-speech tagger that, for each word, classifies it in one of several categories (e.g. nouns, verbs, adverbs, etc.)
- Several partial parsers, that identify Complex Quantifiers, Noun Phrases, and Verb Phrases.
- A partial parser that finds subject-verb and verb-object relationships.
- A module that separates the text sections and chapters.

Each of these modules adds annotations to the texts in the form of XML entities and attributes.

Term extraction

After the linguistic processing, there are some modules that locate several different entities. Some of these components are optional specialists for restricted kinds of terminology, such as dates, scientific names, or unknown domain-specific words. They use various technologies, including regular expressions and language identification technologies.

The unknown domain-specific terms that have a high frequency of appearance in the documents are automatically considered relevant terms, and there will be special hypertext pages for describing them in the generated adaptive web site.

Term classification

Finally, there is a module that classifies the unknown terms, automatically, in a conceptual semantic network, so some meaning can be inferred from their position in the network.

The output of the off-line processing is the following:

- The original documents are returned with XML annotations, which include linguistic analyses of the texts; temporal expressions, and the relevant terms found.
- The conceptual semantic network is returned extended with new concepts, corresponding to the relevant terms that were identified by the Term Extraction module.
- A database is created about these documents, containing information about how to order chronologically the events told in the text; about all the places in the documents where each of the relevant terms appears; and an empty table in which the user profiles will be stored.

This modular architecture allows the author of the adaptive web site to *plug in* optional modules specialist in identifying some particular kind of terminology, such as dates or scientific names.

1.4.2 On-line processing

An **on-line** processing is performed whenever a user enters the system to read the information. Via the user profile manager, users can register and create their profile; they can choose some of the options according to their needs, and specify their interests; and some parts of their profile will be created dynamically, using tests.

After registering, it is possible to browse the generated pages. If the users have time constraints, the documents will be presented to them summarised. It is also possible to ask the system to show summary pages with information from the Internet, which has been automatically collected and filtered.

Every one of the described steps is completely automatic: it is possible to generate and use the whole site just by executing a shell script, and there is no need for supervision. On the other hand, for experienced users and web site authors, the output of each processing module is written in XML files, databases and configuration files, and it is also possible to perform a manual revision of the module outputs and to correct them if necessary.

User profile manager

The generated web pages are not complicated to use, but users are expected to be familiar with browsing through hypermedia, choosing hyperlinks and filling HTML forms. When registering, a new user is asked to complete a few forms in which the system constructs the profile, with some of the following information:

1. The **amount of information** they are willing to read. They may indicate it in various ways, such as the total number of words that the generated web site must contain; a compression rate to be performed to all the web pages; or the amount of time that they want to spend reading the whole site. In some cases, there will be no need of condensing the information, if a user has a large availability of time or asks for a very high number of words. On the contrary, in other occasions it will be necessary to reduce the contents of the site.

2. In the case that a user indicates the amount of time available, it will be necessary to know his or her **reading speed**, measured in words per minute. The new users receive a text, and the system measures how long it takes to each of them to read it all through. Next, their **reading comprehension level** is calculated with a simple test.

3. **Topic preferences**, which can be selected from a set of pre-defined *stereotypes*, or generated specifically by the user by classifying some texts as relevant or irrelevant.

Summary generator

When a user has time restrictions, then the adaptive hypermedia site should provide the information in a condensed way, using some kind of summarisation algorithm. However, the user should always be able to expand a summarised fragment in order to read it all, in case that a fragment is found very interesting and the user does not want to miss any information about it.

User tracker

Although the users specify, at registration time, their interests, it is possible that a stereotype does not describe exactly the interests and the goals sought. Therefore, it is necessary to provide a way in which the user can return feedback to WELKIN about whether the decisions of highlighting some information or eliminating other data were correctly taken. The aim of the user tracker component is to update the user model of interests while the user browses the adaptive site, according to the indications received. This changes on the model take effect immediately, so the page generated right after an update has to take them into account.

Internet retriever

When a domain-specific term has been considered important to the user, it should be possible to retrieve additional data from the Internet concerning that term, so the original information about it can be extended. For performing the search on Internet, context words can be used to improve the quality of the information gotten:

- As keywords, to guide the search using a standard search engine such as `Google` (http://www.google.com) or `Altavista` (http://altavista.digital.com).

- As filter words, to retain only the most relevant of the web pages that have been provided by the previous searches.

The retrieved web pages are put together in a single page that summarises their information.

User Interface

The first purpose of the user interface is to allow new users to register into the system, and to initialise their personal profiles. The reading speed and reading comprehension, if necessary, will be acquired with a test, and the remaining information, availability of time and user interests, will be asked directly to them during registration time.

The pages from the on-line information system have to support the following functionalities:

- Concerning *adaptive contents*, texts are provided summarised if there are time constraints; however, it is always possible for the user to ask for the original contents of the text, as they were before the summarisation process. The information collected from the documents used in the off-line processing is expected to be highly reliable, because it was provided by the user; on the other hand, the information taken from Internet might not be correct. Therefore, it must be always possible to identify the source of the different information. Finally, it must be always possible for the user to indicate the system whether the provided information is or is not interesting, in order to train it and to get better information afterwards.

- Concerning *adaptive navigation support*, the following links are automatically generated: links for navigating chronological information, to read the events chronologically ordered; links to descriptions of semantically tagged new concepts (to read information about persons, locations or artifacts identified in the original documents); links to read sections linearly, as they were written in the original documents; and links to extra information collected from the Internet.

1.5 Textual resources used

This section describes the source texts used for building some example sites, and the different corpora that have been used, at one point or other of the research, for training and testing purposes. Finally, for each module, there are some tables that indicate which are the sections in this thesis where they are described, and which publications are available about them.

1.5.1 Web sites built

Although the architecture described here has been designed in a general-purpose way, for implementation purposes some of the modules were built with a few additional restrictions. The following are the restrictions which affect directly the kinds of texts that can be chosen to generate adaptive sites about them:

- The processed texts must be written in English. All the tools for linguistic processing of the texts have been developed for this language, so it is a requirement that the source texts must satisfy.

- The identification and classification of unknown terms in texts has been restricted to *physical entities*. These include animals, plants, people, artifacts, locations and bodies of water. Terms that refer to

Step	Module	Training	Testing
off-line	Linguistic Processing	Penn Treebank II (WSJ)	WSJ
	Term Identification	WN 1.7	LOTR, The Iliad
	Term Classification	WN 1.7	LOTR, The Iliad
	- Time Expressions		WN 1.7, LOTR, Penn Treebank II
	- Scientific Names		The Voyages of the Beagle
on-line	Summary Generator		The Voyages of the Beagle, DUC 2003
	Internet Retriever		WN 1.7, The Voyages of the Beagle

Table 1.1: Corpora used either for training or testing different modules of the system.

abstractions or actions have not been studied although, in theory, the same techniques could be applied to classify them into semantic lexicons. Therefore, the texts chosen to generate adaptive web sites with WELKIN should not contain a high amount of abstractions if the classification of unknown terms is to work.

Secondly, the purpose of the system is to select a subset of a source text with the information that can be interesting for a particular user. Therefore, for evaluation purposes, it would be desirable to process texts that can be studied from many points of view. In this way, many kinds of interest profiles could be defined by the users, and they could evaluate the adequacy of the information selected by the system to their interests.

Three adaptive web sites have been developed from texts that satisfy these requirements. The three of them deal partly with history, although from different points of view: Darwin's *The Voyages of the Beagle*[3], Osler's *The evolution of modern medicine*[4], and Hegel's *Lectures on the history of philosophy*[5].

These texts are not only either written or translated into English, but also they are smoothly written in correct English, which facilitates the work of the syntactic analysers. The fact that they deal with historical events (the voyages of the Beagle, or historical accounts of different disciplines) implies that they contain relevant terms which are physical entities (people, locations, artifacts...). Finally, they can be studied from different points of view, as they probably contain fragments about historical events, descriptions of the places where those events take place, or accounts of different sub-divisions of the disciplines. As will be seen, the generation of a whole site from the text does not take a long time, so the sites can be put up-to-date with a certain frequency if the source text varies with time (e.g. newswire articles about some topic of interest).

1.5.2 Training and test material

Apart from the above-mentioned material, there are other texts that have been used for the evaluation of small modules in the system, displayed in Table 1.1:

- The *Penn Treebank II* [Marcus et al., 1993] is a corpus that contains a set of texts fully parsed. In particular, it includes the *Wall Street Journal* (WSJ) corpus, a collection of newswire articles from the WSJ newspaper, which has been used for training several linguistic processing applications.

- WordNet [Miller, 1995] is a lexical semantic network, in which words are related to each other with semantic relationships. It has been used as training, in the *Term Identification*, to distinguish which of the new terms are instances and which are concepts, and in the *Term Classification* step. It has also

[3]Obtained from the Gutenberg project, `http://promo.net/pg/`
[4]Obtained from the Gutenberg project.
[5]Obtained from `http://www.class.uidaho.edu/mickelsen/ToC/Hegel-Hist of Phil.htm`

Step	Module	Section	Published as
general	General Architecture		[Alfonseca and Rodríguez, 2002]
off-line	Linguistic Processing	B.1	[Alfonseca, 2000]
			[Manandhar and Alfonseca, 2000]
	Knowledge Base Structure	B.2	[Alfonseca, 2002]
	Term Identification	5.4.1, B.2.2	[Alfonseca and Manandhar, 2002a]
	Term Classification	5 and 6	[Alfonseca and Manandhar, 2002f]
			[Alfonseca and Manandhar, 2002e]
			[Alfonseca and Manandhar, 2002d]
			[Alfonseca and Manandhar, 2002b]
	Time Expressions	6.3	[Alfonseca and Manandhar, 2002c]
	Scientific Names	6.4	
on-line	User profiles	8.1	[Alfonseca and Rodríguez, 2003d]
	Summary Generator	8.2	[Alfonseca and Rodríguez, 2003c]
	User Interface	8.3	[Alfonseca and Rodríguez, 2003b]
	Internet Retriever	8.4	[Alfonseca and Rodríguez, 2003a]

Table 1.2: Sections that describe the main modules, and papers about them.

been used as an additional resource for the identification of temporal expressions and for the collection of new information from the World Wide Web. WordNet will be described in more detail in Section 3.3.

- *The Lord of the Rings* and *The Iliad* are two narrative texts that have been used in order to test the performance of the Term Classification module when classifying unknown physical entities. The choice of two mythological texts is due to the fact that they include several terms that refer to races of people or animals, as well as rare artifacts and places, which can be used to calculate the accuracy of the classification algorithm for physical entities.
- The test collection from the *Document Understanding Conference* is a set of texts which has been used in an international competition of text summarisation algorithms.
- Finally, Darwin's *The Voyages of the Beagle* was also used to test some of the modules, such as the optional scientific names recogniser (an optional sub-module of the Term Identification step) and several components from the on-line processing step.

Table 1.2 shows the chapters and sections in which the most relevant modules are described, and the work that have been published about them.

1.6 Contributions of the thesis

The problem of automatically finding relevant information in large repositories of data, and showing it in a way dependent on the user's interests and goals, is a hard issue that has not been fully solved yet; this thesis aims at advancing a few steps toward that final aim. While designing the framework, special interest has been taken in the two tasks for producing adaptive hypermedia sites: the *off-line processing*, when knowledge is captured, and the contents of the site are obtained and structured; and the *on-line processing*, which deals with the adaptation rules to the users that connect to the system.

The off-line step in the architecture that has been designed combines ideas from several fields, such as Adaptive Hypermedia, User Modelling, and many subfields from Natural Language Processing such as

summarisation systems and conceptual ontologies. The architecture proposed has been implemented as a modular pipeline, where any module can be interchanged by any other that follows the same annotation guidelines, and the remaining components can be left untouched.

In the step of term identification and classification, a new algorithm has been devised for introducing new terms in conceptual semantic networks, and it has been evaluated with WordNet 1.7 [Miller, 1995], which is a large network that contains 74,487 nominal nodes. Secondly, a unified framework has been proposed for being able to compare this algorithm to alternative approaches that may appear. This is also relevant, because previous approaches for learning the meaning of new concepts in an unsupervised way are scant and they all use different training and test corpora, so up to now they have not been directly comparable.

Concerning the recollection of information from the Internet, a new algorithm has been designed for filtering the results obtained with conventional search engines in order to collect information that is more relevant according to the topic of the hypermedia site. The information found in the texts that are used as a starting point for creating the hypermedia site is used in order to retain only the information from the Internet in which the search keywords are used in the same senses than in the original text documents.

Concerning the on-line processing, when a particular user connects the system, a new summarisation algorithm based on genetic algorithms has been devised in order to put together all the collected information in a readable way, taking into account the user's interests.

Chapter 10 describes in more detail the contributions of the whole work and each of its components.

1.7 Thesis overview

In order to fulfil the purpose of this thesis it has been necessary to treat very different fields, such as Term Classification, automatic generation of summaries and User Modelling, each of which has different idiosyncrasies and solutions. Considering that a very large description of state-of-the-art in all these fields at the beginning of the thesis might be unappealing for the reader, the literature review has been divided according to the subject of each part in which the thesis is structured. This is the reason why parts II and III include, independently, literature reviews related to their particular topics, descriptions of the new work presented in this research, and the results obtained.

The thesis has been divided into four parts. The first one contains an introduction with the motivations, purposes and a high-level design of the framework for **automatically creating adaptive hypermedia web sites**, as a whole. Secondly, Part II describes the first step in creating adaptive hypermedia systems, the **off-line processing** in which contents are collected, and the knowledge represented in those textual contents is formalised as a conceptual semantic network. Part III describes the work on the **on-line processing**, when users access the system and it has to adapt the contents to their interests and needs. Each of the components of the framework is evaluated independently, and, in Part IV, a global evaluation of the whole system is presented.

Each chapter starts with an introduction about the topics that will be addressed in the chapter, and a small paragraph describing its internal structure. Also, every chapter ends with a brief summary that condensates its contents in a few paragraphs and, when judged necessary (e.g. when there was original work described in the chapter), a brief discussion with some of the preliminary conclusions that can be drawn.

Part I

- **Chapter 2** reviews the state-of-the-art of hypermedia generation, adaptive hypermedia and user modelling. Special emphasis is set on the application of Natural Language Processing techniques for Adaptive Hypermedia generation.

Part II

The **off-line processing** centres on the identification and classification of the relevant concepts and sections in the original text. In this step, a database is created with all the information extracted from the content texts. The chapters included in part II are the following:

- **Chapter 3** contains a literature review about automatic lexical knowledge acquisition: acquisition of new words and their meaning.
- **Chapter 4** describes the Distributional Semantics hypothesis, which will be extensively used in the whole research. Different distance metrics between the meaning of concepts are discussed, and an empirical justification of the hypothesis is provided.
- **Chapter 5** describes the approaches followed for extending conceptual semantic networks with new terminology learnt from the domain-specific documents from which the hypermedia site is produced, using Distributional Semantics techniques. This includes the Term Identification and the Term Classification steps.
- **Chapter 6** describes other approaches that have been used, outside the scope of Distributional Semantics, mainly with regular expressions and other kinds of patterns, for automatic identification and classification of general terminology, and particular kinds of entities such as dates.

Part III

The **on-line processing** includes the work that is done interactively with the hypermedia site visitor, when a petition of information arrives. Part III describes the components of the system that, using the information from the database described in Part II, and the information available about the user, constructs on-the-fly the hypermedia pages and the links to other pages and provides them to the user.

- **Chapter 7** describes the state-of-the-art techniques in text summarisation, either from a single-document or from multiple documents, and cites some work in applying summarisation to hypermedia generation.
- **Chapter 8** describes the modules that generate the final hypertext pages in the WELKIN implementation, depending on the users' profiles, and add the hyperlinks that structure the generated site. This chapter describes the different features of the users that are modelled in the system and how they affect the presentation of contents. A new summarisation algorithm is described here and evaluated. This chapter describes, as well, the modules that collect information from Internet in order to complete the hypermedia pages with additional data.

Part IV

- **Chapter 9** contains the usage evaluation of the system. Three case studies are described separately, corresponding to three adaptive hypermedia sites that were generated using the system. Finally, this chapter describes a controlled experiment with users.

- **Chapter 10** contains the conclusions or the work: its contributions, a comparison with previous work on the field; and the open lines for future work, either in general and separately for each one of the components in which the architecture is divided.

Appendixes

Finally, there are a several appendixes that contain other information that is relevant, but not central, for the purposes of the thesis:

- **Appendix A** lists all the abbreviations used in the thesis.
- **Appendix B** describes all the additional work that was necessary for the work, such as all the linguistic tools needed to process the texts, the design of some of the databases, etc.
- **Appendix C** shows several examples of texts generated by the different modules for content adaptation: the selection of relevant information to the user, the output of the summariser, and the information collected automatically from the Internet.
- **Appendixes D and E** end the thesis with the translation into Spanish of the Introduction and the Conclusions of the thesis.

Chapter 2

Adaptive Hypermedia

Adaptive Hypermedia (AH) studies the techniques and methods for combining hypermedia with user modelling, in order to generate hypermedia systems tailored to the needs of each user. It is a somewhat recent area, motivated by several factors. On the one hand, the large size of modern storage devices and the largeness of the Internet can very easily disorient the users and make it difficult for them to find the information that they need. This produced the appearance of many kinds of systems that help the user navigate through hypermedia, such as Information Retrieval engines that search for information that appears relevant according to a user's query. On the other hand, hypermedia systems allowed for a user's interaction much richer than printed material can offer. These possibilities of interaction motivated the appearance of educational systems, on-line information systems, and many other kinds of applications. It was short before User Modelling techniques started to be applied to hypermedia systems.

Partly due to the fact that the area is recent, and partly due to the many possible applications of AH, it happens that in Adaptive Hypermedia there is a big diversity of solutions for each adaptation problem and no standard framework has appeared that unifies the most relevant advances in the field. This chapter provides an overview of current work in AH, from different perspectives, in Section 2.1. Next, Section 2.2 describes the particularities of AH systems that use Natural Language Processing techniques.

The above mentioned reviews pay special attention to AH systems that use the Internet as the mean to convey the information, as well as standard web browsers for the interface. It is worth mentioning that most of the systems recently described follow this approach.

2.1 Adaptive Hypermedia review

There are many different features that can be used to study and classify AH systems. Some comprehensive reviews are those written or co-authored by Brusilovsky [Brusilovsky, 1996, Brusilovsky et al., 1998, Brusilovsky, 2001], which describe four different dimensions:

- Applications: *where can it be used?*
- User characteristics: *adapting to what?*
- Hypermedia elements that are adapted: *what can be adapted?*
- Techniques for adaptation: *how can the adaptation be done?*

Jameson [1998] proposes other dimensions of analysis, such as the input for user model acquisition, and the empirical foundations of the adaptation techniques. Other comprehensive review is that found in Carro

[2001] which focuses on AH applied to Hypermedia Education, and adds six new dimensions that can be used to classify general AH systems:

- Degree of generality: whether the system allows for many kinds of applications to be built with it.
- Design of the Hyperspace: the topology of the graph formed with the hypermedia documents and the links between them.
- Access to the information: local or remote.
- Ways to trace the user actions.
- Maintenance of the systems.
- Technological requisites.

This review will consider only some of the previously mentioned criteria. The remaining features, that are not studied explicitly, will be seen implicitly with some of the features selected. This is the case of the input for the user model and the ways to trace the user actions, which can be reviewed together with the user characteristics. On the other hand, features such as the degree of generality, the access to the information, maintenance and the technological requisites apply to any software application, not only to AH systems, and they have not been taken explicitly into consideration. Some reflections about these topics can be found in the conclusions, in Chapter 10.

In summary, the following classification criteria will be considered:

- Applications (section 2.1.1).
- User characteristics (section 2.1.2).
- Hypermedia elements that are adapted and techniques for adaptation: because the techniques are different for each kind of element that can be adapted, these two features can be studied together (section 2.1.3).
- Design of the hyperspace (section 2.1.4).

This review does not intend to include a list of every related work in the area, but to provide an insight into the possibilities of Adaptive Hypermedia: its applications, methods and techniques. For a more detailed list of related work, please refer to the above-mentioned reviews and the report by Kobsa et al. [1999].

2.1.1 Applications of Adaptive Hypermedia Systems

Brusilovsky [2001] identifies six different areas of application of adaptive hypermedia systems: *educational hypermedia, on-line information systems, information retrieval, systems for managing personalised views in information spaces, on-line help systems* and *institutional hypermedia systems.*

Educational systems have traditionally provided the same material to all students, independently of their profiles. The writer of a textbook usually decides which is the best learning strategy for an average student, and everyone has to follow it. However, literature in psychology affirms that people have different approaches to learning and studying [Dunn and Dunn, 1978]. **Adaptive Educational Hypermedia** appeared in order to adapt the contents of the course to the varying needs of the students. Characteristics that can be modelled are the user's age, language or preferred learning style. A feature that is very important in educational systems is the user's previous knowledge. A web page can be *unclear* for a student that does not have the necessary background, *interesting* for a student that has just learnt the necessary pre-requisites to understand it, and *boring* for a student that already knew its contents beforehand. Adaptive educational hypermedia should keep track of how the users' knowledge vary, by updating their profiles as they read web pages or solve exercises, and use that profile to suggest what to visit afterwards.

Some educational systems are ANATOM-TUTOR [Beaumont, 1994], ELM-ART [Weber and Specht, 1997], CAMELEON [Laroussi and Benahmed, 1998], TANGOW [Carro et al., 1999], Arthur [Gilbert and Han, 1999], AHA [de Bra et al., 2002], ACE [Specht and Oppermann, 1998] or HEZINET [López-Cuadrado et al., 2002], amongst many others.

On-line Information Systems comprise the systems whose aim is to provide access to some information. It is a broad class that includes systems as heterogeneous as web sites describing some specific domain of knowledge, electronic encyclopaedias, information kiosks, virtual museums, handheld guides, e-commerce systems or performance support systems. Different users may have different knowledge and interests, and therefore the contents presented to them cannot be the same; in particular, the system should help the users in finding the information they need. Each kind of system has different possibilities of adaptation: electronic encyclopaedias can keep track of which are the concepts that the user knows in order to suggest similar or related concepts [Hirashima et al., 1998], or in order to describe a new concept by comparing it with the ones already seen [Milosavljevic, 1998]; kiosks [Fink et al., 1998] can take advantage of the place where they are located to show the user relevant information; virtual museums [Milosavljevic et al., 1998, Not et al., 1998, Oberlander et al., 1998] and handheld guides can take advantage of the location of the user in order to narrate context-dependent information; e-commerce systems usually model users' tastes with the products they have visited, bought or rated, in order to make user-adapted suggestions [Domingue et al., 2002]; and performance support systems, such as expert systems, help the user in attaining a goal, such as providing medical treatment, so these systems have to be aware of the user's situation and needs.

On-line help systems is a different kind of AH systems, but they can take advantage of the same techniques that are used by on-line information systems such as virtual museums. In a virtual museum, it is the user's context, defined by the geographical locations, which determines the explanations provided. In the same way, in help systems the context consists of the last actions performed in some computer application. This context can be used to infer which is the information that the user needs.

Information Retrieval (IR) [Baeza-Yates and Ribeiro-Neto, 1999] is the task of finding information that is relevant to the user in huge repositories of data. The user need is usually stated with a query, either as a natural language question, a set of keywords, an expression with logical operators, or a set of documents that the user finds interesting. Research in this area was encouraged by the Text REtrieval Conferences (TRECs) [Voorhees and Harman, 2001], organised now for more than ten years by the National Institute of Standards and Technology (NIST), a competition in which the participant systems compete in locating relevant information. The field has thriven since the advent of the Internet, due to the large amounts of information available, in which users can easily feel at a loss to find the data they need. An IR system can use hypermedia techniques, such as allowing the user to browse the results, and adaptive techniques, such as showing a small summary of the contents of the retrieved web pages, adapted to the user's interests.

There are other systems which are closely related to IR. The so-called **systems for managing personalised views** can be considered a particular case of IR, where the user defines a *personalised view* by means of one or several goals or interests, and the Internet hyperspace is reduced to include only the documents that are relevant according to that view and the hyperlinks amongst them. **Institutional Information Systems** allow users to find information inside an institution (e.g. in an intranet). It can be considered a particular case of IR, and it is perhaps more flexible because the employees of the institution may be in contact with the developers and ask for personalised functionalities. **Document Routing** (DR) [Harman, 1991] systems inform all the users about the new documents that have entered the system and might interest them. To do that, they have static user profiles with their general interests. We shall return to IR in

Section 3.2.2.

2.1.2 Characteristics to which the systems adapt

According to Brusilovsky [2001], we can distinguish two sets of features to which a system can adapt: the user's personal characteristics and the environment. The user's personal data, again, can be divided into two separate groups [Kobsa et al., 1999]: the features that are part of the user, such as his or her age and knowledge, and the *usage data*, or the particular ways in which a user interacts with the system.

User's features include the following:

- *Individual traits*: these are static characteristics of a user: they are not supposed to vary, at least in the short term. They can include the user's date of birth, language and geographical data, amongst many others.

  These features are relatively static in time. Therefore, the system can obtain them by asking the users when they register for the first time in the system, and then it can store them in the users' profiles. However, some features, such as the psychological traits, have to be acquired using psychological tests. For example, some educational systems use tests to know which is the learning style preferred by each student [Laroussi and Benahmed, 1998] [Carver et al., 1996] [Paredes and Rodríguez, 2002]. The user's availability of time is taken into account by other systems, in order to provide longer or shorter explanations [Milosavljevic et al., 1998]; a similar feature for written text is the user's reading speed, taken into account by Ng et al. [2001] in order to infer whether the reader is reading with attention, skipping text or leaving the application unattended.

- *Previous Knowledge*: awareness of the user's previous knowledge and beliefs is usually required in order to provide useful explanations. Information support systems should not repeat information that has been already sáid so as not to bore the user [Milosavljevic et al., 1998, Oberlander et al., 1998]. Also, the user's previous knowledge can be used to make comparisons between a new topic and those that the user already knows [Milosavljevic, 1998], or to provide evidence against a user's false belief [Zukerman and McConachy, 1993]. Educational systems also model user knowledge, because some sections may be unclear if a student has not learnt other sections that provide the necessary background [Weber and Specht, 1997] [Carro et al., 1999] [Gilbert and Han, 1999] [de Bra and Calvi, 1998] [Specht and Oppermann, 1998].

  Knowledge can be represented in AH systems as a semantic network [Brusilovsky, 1996]. The main concepts provided by the hypermedia system can be related to each other, in the network, by means of one of several relationships. For example, *precondition* can relate two nodes such that knowledge about the first one is necessary in order to understand the other; or *part-of* can relate one node with others, if these other nodes decompose the knowledge represented by the first one. The users' previous knowledge can be either directly asked to them, when registering; constructed by remembering which are the web pages that they have already either visited, studied or passed; or inferred from tests they have done.

- *Experience with navigation*, which may determine the complexity of the generated web pages: a system may show to experienced users more hyperlinks and navigation options than to novices.

- *Interests and goals*: user's interests are vital for any adaptive systems to perform satisfactorily. IR and DR systems traditionally stored this feature as bookmarks of web pages for which the user expressed strong interest; therefore, hypermedia pages that are similar to those bookmarked by the user can be assumed to be relevant. Other ways in which user interests can be recorded are by explicitly asking the user in an initial interview [Fink et al., 1998] or with a history of the kind of topics that the user has viewed. This last procedure has the advantage that the system can track if the user's interest shifts from one topic to other by examining the sequence of pages visited [Hirashima et al., 1998]. This feature is central in *recommender systems*, which attempt to recommend the user some product or service.

- *System-specific features*: some AH environments require specific information. For example, medical diagnosis systems need to record the user's symptoms, their evolution with time, their habits and the drugs that they have been medicated previously [Carenini et al., 1994, DiMarco et al., 1997].

Sometimes, user profiles are stored as *stereotypes*: template profiles that contain the combinations of user features that are considered more common. This can be done when there is a set of stereotypes that can be expected to represent the whole population of users (or, at least, the majority of them). For example, if the user profile determines the complexity of the page contents, there may be stereotypes such as *novice*, *beginner*, *intermediate* or *expert*, and the system will only expect four kinds of users. Stereotypes are simpler to manage and therefore adaptive systems based on them should be more maintainable, but sometimes it is necessary to have a finer-grained description of the user. But, even in this case, stereotypes can be useful. Given that it is difficult to initialise some characteristics of the user profiles, some systems opt for initialising new users with stereotypes, and next these profiles are used and updated as the user interacts with the system. This was the approach taken for systems such as those described by Beaumont [1994], Carro et al. [1999] and Oberlander et al. [1998].

Environment features such as the software and the hardware used by the users can determine the types of contents that will be shown to them. For instance, if the hardware is old and slow, or if the band-width is low, it may be preferable a still snapshot than a motion picture; if the screen is small, the amount of text might be reduced, or the images might be eliminated (e.g. in a mobile phone or a Palm Pilot). Environment features are also used by *advertising systems* in order to provide advertisements of products and services that are available at the geographical location of the user.

2.1.3 Elements that are adapted

Hypermedia consists of pages and links that connect them. A page usually includes some multimedia contents (text, images, audio, video, etc.) and links to other pages. Therefore, there are two kinds of adaptations that can be performed: on the contents of the pages and on the navigation options. Figure 2.1 shows a taxonomy of the different adaptation techniques that can be performed. The remainder of this section briefly describes each of them.

Adaptive presentation

The adaptation of the page contents, usually called *adaptive presentation*, consists in showing different information to different users according to their profiles. The changes may consist in providing the same textual information in different ways (e.g. in different languages, summarised, with different writing style,

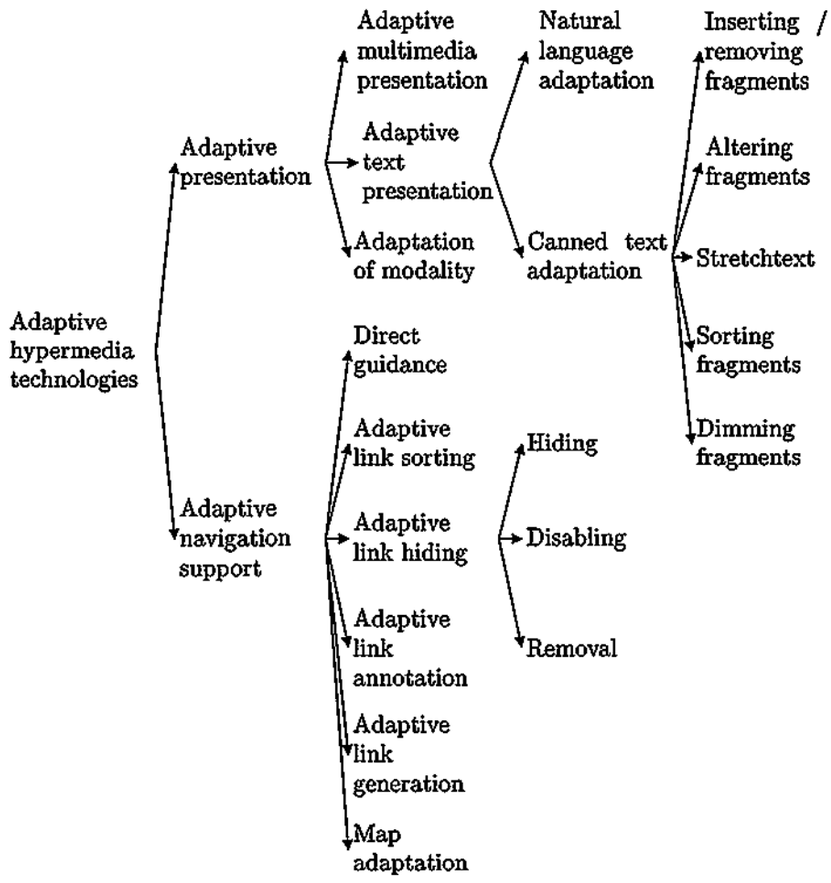


Figure 2.1: Taxonomy of adaptive hypermedia technologies, from Brusilovsky [2001].

more or less in-depth, etc.) [Carro et al., 2002, Oberlander et al., 1998]; changing the textual information depending on the user interests (e.g. the output of an IR system to two different user queries); or varying the multimedia elements in the page (e.g. depending on the size of the screen or speed of the network connection). Several techniques are described in the next paragraphs.

Adaptive text presentation Text is still the media that is more used on the Internet, and it is still very easy to handle by adaptive systems, as portions of a long text can be easily deleted; whereas deleting a portion of a movie or modifying a picture requires more complex software. Therefore, adaptation methods and techniques for text were the first that were used, and they have been more studied than techniques for other media elements.

Canned text adaptation is the simplest form of adaptive text presentation, but still it is sufficient for many applications. If a user is a novice in a field of knowledge, it may be necessary to include basic explanations for him or her to understand an advanced text; while those explanations would be unnecessary and boring for a an expert in the field, who might want to read more advanced passages. The system may have additional explanations both for novice and advanced users, that will be added to the contents of the hypermedia page. In the same way, by modifying the text it is possible to include comparative explanations relating a concept with others studied previously, if the system knows that the user has seen them before.

Secondly, the system can store alternative fragments of text with the same information, but intended for different users. For example, the same text can be written in different languages; or it can be written with technical style for an adult user and with familiar style for a child.

Finally, the system can also sort several fragments using some metric of relevancy to users.

The following are the most common operations that can be used with fragments of texts:

- Removing or adding fragments. In this approach, a page is usually composed of a list of textual fragments, having each of them some preconditions about the degree of knowledge or other user features that a particular user should satisfy. When a hypermedia page is generated, the preconditions of each fragment are evaluated, and the system selects the fragments whose preconditions hold.

- Altering fragments. In this approach, the system has several versions of the pages or fragments, and it selects the version which is adequate to the user characteristics (expertise, age, language, etc.). This technique is particularly appropriate if the system uses stereotypes [Carro, 2001].

- *Stretchtext* is a special technique consisting in that the document contains one or several highlighted hotwords or sentences, and when the user chooses one of them it is replaced with a fragment of text. The system initially decides which fragments are expanded and which are collapsed, but the user can modify it by selecting the hotwords to collapse a shown fragment or to show a hidden fragment [Boyle and Encarnacion, 1994].

- Sorting fragments consists in showing the fragments in order of relevance to the user. This technique is very used in IR systems, when the hits of the user's query are returned ordered.

- Dimming fragments, as described by Hothi and Hall [1998] can be done by showing them in a colour similar to the background, so the fragments that are not dimmed receive more attention. Hothi and Hall [1998] report that most of the users agreed that this method reduced information overload, while at the same time the fragments can be read if a user is interested. This has the advantage that if the

system is not sure whether a fragment will be of interest or not, it can show it dimmed, and it is the user who decides whether to read it or not.

Although canned texts can serve the purposes of many adaptive hypermedia systems, natural language generation and understanding is one of the trends in content adaptation [Zukerman and Litman, 2001]. Applications include adaptive narration (producing texts according to the user's interests and goals) [Oberlander et al., 1998, Milosavljevic et al., 1998], adaptive guidance (guiding the user to achieve a goal, e.g. information kiosks that guide tourists to monuments and hotels) [Geldof, 2000, 1998] and adaptive comparison (teaching a user the differences between an unknown concept and the known concepts) [Milosavljevic, 1998]. Techniques for combining AH with Natural Language Generation techniques will be explained in more detail below in this chapter.

Adaptive multimedia presentation Modern hypermedia systems have the possibility of displaying different types of media, such as text, images, music, speech, video, interactive controls, etc. Often, fragments of different media have the same content, and the system chooses which is the most appropriate for each user. In other cases, the system can present some of them at the same time, such as text and speech. When the system presents the same contents but with different media (e.g. with a video, an image or text) then the adaptation is called *change of modality* or **adaptation of modality.**

Different methods for adapting the media to the students have been proposed by Carver et al. [1996], Fink et al. [1998] and Specht and Oppermann [1998].

Adaptive navigation support

Adaptive navigation support consists in adapting the hyperlinks that allow the users to navigate in a hypermedia site according to their profiles. The motivation for adapting the links is that of helping the users to find the information they need, by leading them to the web page they are looking for. This can be done by many techniques, such as annotating the links with a summary or an extract of what the user will find, as most IR systems do, or with colour codes (e.g. different colours for recommended and non-recommended links); by hiding the links that the system believes will not be useful for a particular user, etc. The following paragraphs describe several adaptation techniques that can be performed.

Direct guidance This technique consists in suggesting a *next* link to the user, in function of the features contained in the profile. Therefore, the user does not need to decide between a multitude of links but can always follow the recommendation of the system.

Adaptive link sorting This consists in sorting the links according to their relevance. By using the user's interests or goals, the links whose contents are more related to the user interests (e.g. expressed with a query for an IR engine) or the links that lead the user nearer a goal (e.g. in decision-support systems) shall appear at the beginning of the list. It must be noted that, in other applications, such as educational systems or information support systems, the continuous change of order of the links might disorient the user, and thus this technique is not always recommended.

Adaptive link hiding Web site designers usually have a single entry page, from which they have to ensure easy connectivity to all the contents of the site. These *"portal"* pages usually contain a large number of links

of which every user is interested in just a few. For a novice user it can be difficult to decide which are the ones that lead to the required information.

This technique consists in selectively removing the links depending on the user. There are many criteria that can be applied. In some cases, the system deems that a link leads to information on which the user is not interested. On the other hand, most hypermedia education systems restrict the access to some complex sections until the user has proved expertise on the simple ones.

The deletion of the link can possibly involve deletion of the text that contains the link, if it does not produce incoherence in the web page.

Adaptive link annotation This consists in adding some information to the links so the user can guess some characteristics of the information to which those links lead. The annotations can be text (e.g. a summary or an extract of the web page pointed from that link), icons, or colour codes, which may indicate, for example, whether the system recommends a link or not.

Adaptive link generation In some cases, the profile of the potential users is very fine-grained, allowing many possible combinations of features, and it would be desirable to allow the system to create the links on the fly, while showing a page to the users, instead of designing them all at authoring time. We can distinguish three kinds of techniques in this group: discovering new useful links that will be permanently added to the hypermedia system (e.g. and information-support system that locates on the Internet a useful link and adds it permanently to a *links* page); generating links for similarity-based navigation between pages; and dynamic recommendations of relevant links depending on the user profile.

Map adaptation Some hypermedia systems provide a graphical view of the link structure of the hyperspace. This can help the user by making it clear which nodes are acting as *index nodes* or *hub nodes*, pointing to many other pages with related information; and which documents are popular, if they are pointed by many pages. These maps can also be adapted according to the user needs. Some examples are those found in the the review by Benford et al. [1999], and the system described by Mukherjea [1999].

2.1.4 Design of the hyperspace

In hypermedia systems there are two different resources that have to be given structure. First of all, the hyperspace is defined with a set of documents and hyperlinks between them, which define the possibilities of navigation between documents. This structure can be described as a directed graph, where the nodes represent the documents, and the arcs are the hyperlinks.

Secondly, hypermedia pages contain information, and that information also has to be structured. A possible way to encode the information is also as a graph, where the nodes are the concepts for which there is some information available, and the arcs are relationships between those concepts.

For designing a hypermedia system, and specially in the case of Adaptive Hypermedia, where some decisions have to be taken about the information presented to the user, it is useful to define a mapping between the hyperspace and the conceptual representation of the information available in the system [Carro, 2001]. It is important to note that the mapping is not necessarily bijective, i.e. a single document may describe more than one concept and a single concept may appear in more than one document.

Therefore, in order to create the hyperspace structure of an adaptive hypermedia system, according to this framework, it is necessary to create the conceptual structure, and to associate to each node the contents or

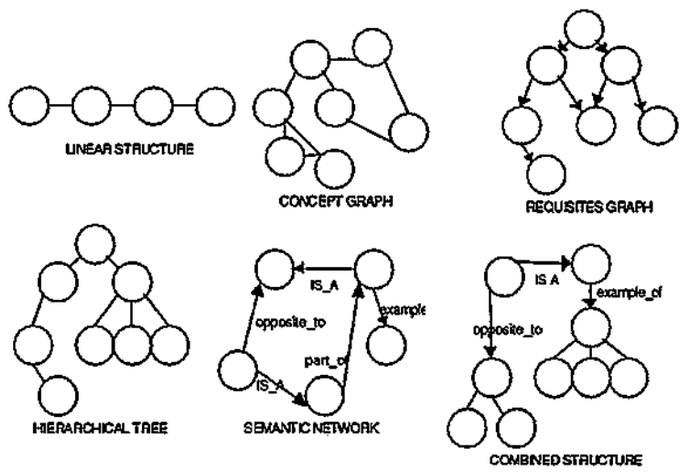


Figure 2.2: Different ways to design the hyperspace in an adaptive hypermedia application, from Carro [2001].

documents that contain information about them. This approach is useful because there is a clear distinction between the structure and the contents, and it is therefore more maintainable and reusable. Figure 2.2 shows several possible structures that can be given to the hyperspaces. The different designs will be described in the remainder of this section.

The simplest structure that can be designed is the **linear** or unidimensional, in which every page points to other two pages, with a *previous* and a *next* link. The only possibility for navigating the hyperspace is sequentially, in the same way that a book is read. This design is not suitable for complex hypermedia applications, given that, usually, a concept is related to more than one other concept, and therefore it would be desirable to allow the user to move, from one node in the hyperspace, to several other nodes.

A different option is that of a **concept graph**, when there exists a bijection between the hyperspace and the concept graph, i.e. when each document represents univocally a concept, and the links between documents represent relations between those concepts. In this case, a hyperlink can be assumed to represent the generic relationship *is related to*, without specifying which is the kind of relation. When the structure is translated to the hyperspace, each node is represented with a single hypertext document, which will be linked to the documents related to it in the concept graph. The user can thus navigate along the concept graph from node to node in any direction. This structure was used in ANATOM-TUTOR [Beaumont, 1994].

There is a family of hypermedia generation systems that take linear text and produce the site by looking for keywords, in the bodies of the different sections, that refer to the titles of other sections. For example, in a hypermedia dictionary, there may be links from words in a definition toward the definition of those words [Raymond and Tompa, 1987, 1988]; or the names of people that appear in a hypertext biography may be linked to their own biographies. The structure of the hyperspace obtained with these procedures is a directed concept graph, where a concept c is said to be related to a concept d if c appears in the page that describes a concept d.

Requisite graphs are a special type of concept graph, in which the only relationship between nodes is a directed relation called *prerequisite*. If a node is a prerequisite of other node, the users cannot access the

second node before they have visited the first node. The set of accessible nodes contains only the nodes such that all their prerequisites have been already seen. This approach was followed in educational systems such as CAMELEON [Laroussi and Benahmed, 1998] and AHM [da Silva et al., 1998], where the relationships between two concepts specify the degree of knowledge required to the students about the first concept in order to allow them to access the second one.

For some applications it is appropriate to structure the information as a **hierarchical tree**. In this case, the hyperspace consists of a set of nodes that are organised via the relationship *is a part of*. With this relationship, it is possible to establish a taxonomy, with the complex nodes located on top of their constitutive parts, and placing at the root of the hierarchy the nodes that are not part of any other node.

Some systems for automatically translating linear books into hypermedia proceed by placing a single root which represents the whole book, with children that represent a chapter each; each chapter will have children that represent a section each, and so forth. This hyperspace can be automatically constructed if the text contains some markup indicating which are the chapter and section titles [Furua et al., 1989]. A widely used tool that follows this approach is *LaTeX2HTML*, which takes a text written in LaTeX and outputs that same text in HTML with hyperlinks that allow the user to move down to subsections, to move up and to move ahead to the next section. It also generates links to footnotes and references.

Therefore, in a hierarchical tree, the more general concepts are located higher in the hierarchy, and the most specific pieces of information are at the bottom. In some cases, when a specific concept is part of several more general concepts (e.g. *conditional probability* might be, at the same time, a subconcept of *probability* and a subconcept of *entropy*), the overall structure is not a tree, but a taxonomy. Hierarchical trees have been used, with some modifications, in educational systems. In ELM-ART [Weber and Specht, 1997], nodes are structured as a tree with the *part of* relation, but each node contains additional information such as pre-requisites; in AHAM [de Bra et al., 1999b, Wu et al., 2000], there is a similar structure where nodes can also store additional information.

Semantic networks are a more sophisticated representation system than concept graphs. A semantic network consists of a set of nodes and a set of relationships between them. As in the case of the concept graph, the nodes represent units of information, but the relations can be of many different kinds, depending on the application needs. For example, a concept can be linked with its *opposite* (e.g. *happiness* to *sadness*), with its generalisations (e.g. *man* to *person*), with its specifications (e.g. *man* to *carpenter*), with its pre-requisites (e.g. *conditional probability* to *probability*), with each of its constitutive parts (e.g. *hand* to *finger*), and virtually any other kind of relationship. This architecture is useful for representing complex knowledge, and was implemented in DCG [Vassileva, 1998].

A particular case of a semantic network is an **activation/inhibition network**. In this case, each node in the network represents a concept or an area of interest, and nodes are related through two kinds of relations: *activation* and *inhibition*. When a user asks for information about some concept, automatically, the concepts that are connected to it through activation arcs are selected (possibly to be included in the generated text), and the concepts in incompatible areas, which are connected to it through inhibition arcs, are excluded [Stock and the Alfresco project team, 1993].

Finally, some systems use combined structures, taking ideas from several of the structures described above. For example, the general concepts might be structured as a semantic network, but then they could be divided into subconcepts organised as hierarchical trees, as in MANIC [Stern, 1997].

2.2 AH with Natural Language Processing Techniques

One of the main goals of Natural Language Processing is to allow a computer to interact with people in the same way that people interact each other [Zukerman and Litman, 2001]. The idea that people use some model of their interlocutor when they communicate is widely accepted by the research community. For example, when talking to two different people one may vary the language, the language style, the intonation, and the presuppositions of the other person's knowledge. Furthermore, people usually update the models of their interlocutors when they interact.

In the case of AH, it can be considered that the user's interaction with the system functions as a dialogue, where the user's utterances are the mouse clicks, understood either as questions or as explicit petitions to change the user's profile —the system's model of the user. The answers of the system, on the other hand, are the generated hypertext pages [Oberlander et al., 1998]. In this interaction, Natural Language Understanding (NLU) techniques can help the system acquire information about a particular domain, or about the users, and Natural Language Generation (NLG) techniques can help generate text that is adequate with respect to the user's needs, providing a higher degree of versatility than a collection of canned text fragments [Zukerman and Litman, 2001].

Many AH systems that use NLG techniques in order to generate the hypertext have a similar high-level architecture. As an example, Figure 2.3 shows the architecture designed by Milosavljevic et al. [1998], applied both to a virtual museum and to a hypermedia encyclopaedia [Milosavljevic, 1998].

Figure 2.3 (a) shows a traditional NLG system. The information that has to be contained in the generated text is usually fed into the system as a *communicative goal*. This may be a user query, such as a request for information about some particular topic; or a user's utterance in a dialogue system. The first module, the **Text Planning Component** takes these goals, and has to generate a *discourse plan*, containing the steps that have to be performed in order to convey the information to the user. In that process, it can obtain information from:

1. An internal Knowledge Base, where it has the information provided by the system (e.g. about a course, a virtual museum, an encyclopedia, etc.), in some internal format, such as a semantic network.
2. The user model, from which it can obtain the user's preferences and interests.
3. A library of discourse plans or scripts, from which a template-plan will be selected.
4. The discourse history, which shall be used so as not to repeat information that was provided to the user before, and in order to make comparisons and references to previously seen topics.

The other module, the **Surface Realisation Component**, takes the discourse plan and generates the appropriate text in natural language. In order to do that, it needs a lexicon in which it finds the words that express the concepts in the discourse plan, a grammar in which it finds how to construct the sentences, and, possibly, a list of cue phrases used to express rhetorical relationships between the sentences.

For example, let us suppose that the user asks an electronic encyclopaedia for information about a *roan*. First, the Text Planning Component looks for and finds, in the plan library, a plan for generating a definition. Looking in the knowledge base, it discovers that a *roan* is a type of *horse*, with the added feature that its colour can be described with the phrase "brownish thickly sprinkled with white or grey". Next, the plan for producing a definition states that the user model and the discourse history must be checked in order to discover whether the user already knows what a horse is; otherwise, some introductory sentences about the concept *horse* have to be added after the definition of *roan*. The list of things that have to be provided to the user is then sent to the Surface Realisation Component, that selects the appropriate words from the lexicon

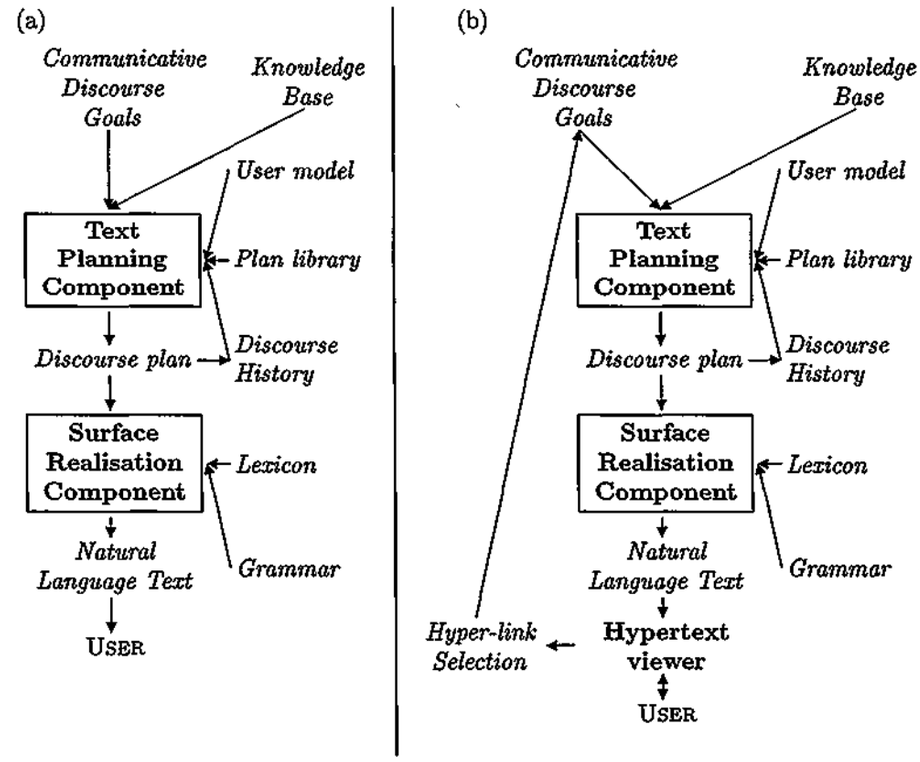


Figure 2.3: Architecture of an adaptive hypermedia system based on Natural Language Generation techniques, from Milosavljevic et al. [1998]. (a) Traditional NLG architecture; (b) dynamic NLG hypertext system.

Element	Semantic net	FOPL
Instance	Node	constant: *Las_meninas, Diego_Velazquez*
Concept	Node	predicate: *painting(Las_meninas), person(Diego_Velazquez)*
Relation	Arc	binary predicate: *made_by(Las_meninas, Diego_Velazquez)*

Table 2.1: Ways in which different items from the knowledge base can be stored in a semantic network (see Figure 2.4) and as a FOPL theory.

and the constructions from the grammar. Different lexicons and grammars can be selected if the users have different languages.

Figure 2.3 (b) describes a looping structure, in which the output of the NLG system is displayed in a hypertext browser, where the user can select a link that is then translated into the next discourse goals. In this architecture, interaction with the system can be considered as a dialogue between the user, which questions the AH software by selecting a link, and the system, which generates the hypertext page as an answer, while taking into consideration the dialogue history.

In order to build a system according to this model, there are three resources that have to be designed:

- The Knowledge Base: how the information is obtained and represented. This is usually done off-line before the users access the system.
- The Planning Component: how the plan is built from the goal and the different resources available: the knowledge base, the user model, the plan history and the dialogue history.
- The Surface Realisation Component: how the plan is translated into natural language.

The following subsections review current NLG-based AH systems according to these dimensions.

2.2.1 Techniques for creating the Knowledge Base

The Knowledge Base (KB) is the place where the system stores all the information that can be provided to different users, in some internal format. Usually, this information includes the following kinds: *concepts*, which refer to kinds of entities about which there is some information in the network (e.g. statue); *instances*, which refer to particular examples of the concepts (e.g. the Statue of Liberty); and *relations*, which specify relations between concepts and between instances and concepts (e.g. the relation is-a-kind-of relates the instance *Statue of Liberty* with the concept *statue*).

There are different ways in which the Knowledge Base can be represented. As an example, Figure 2.4 shows an example of a Knowledge Base structured as a semantic network, and Table 2.1 shows how that same information could be stored as First Order Predicate Logics (FOPL) axioms. In a semantic network, both concepts and instances are stored as nodes (maybe with a flag indicating which nodes are concepts and which are instances), and relations are represented with arcs between the nodes. In FOPL, on the other hand, instances are represented as constants, concepts as unary predicates, and relations as binary predicates.

The KB can be constructed by hand, by consulting experts in the domain that has to be represented. However, if the knowledge base is large, it is not feasible to construct it all by hand and it is necessary to extract at least part of the network automatically. This procedure was applied by both Oberlander et al.

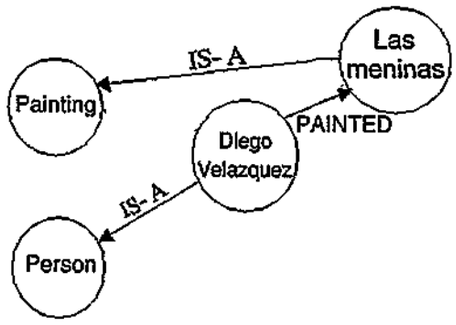


Figure 2.4: Semantic network with concepts, instances, and relations amongst them.

[1998] and Milosavljevic et al. [1998] to translate a museum database containing descriptions of items into internal representations.

The complexity of this automatic translation is, as could be expected, dependent on the format of the source data from which the information shall be extracted.

- If the source data is **structured**, such as relational database indicating for each museum item its date, location, author, style, etc., then the change of format is relatively straightforward.
- When the source data is **semi-structured**, i.e., there is some structure in it, but not completely, then part of the information might be difficult to obtain. For example, Milosavljevic et al. [1998] used a museum database where some of the information, such as the date or location of an item, was stored as separate fields in the entry; but the remaining information was stored in natural language in a slot labelled *observations*, so those last pieces of information were more difficult to analyse.
- Finally, when all the information is **unstructured**, e.g. written in a natural language and not following any guidelines, then it is necessary to use some kind of Natural Language Processing (NLP) techniques in order to analyse and extract the required information for the Knowledge base.

Transforming unstructured text into a Knowledge Base or into hypertext is a difficult task, that requires some level of language processing. There have been previous attempts to transform unrestricted text into hypertext automatically [Hahn and Reimer, 1988] [Clitherow et al., 1989] [Blustein, 1994] [Ragetli, 2001], with different results. Carter [1996] used Information Retrieval and Natural Language Techniques, and found that his context-sensitive hypertext was not acceptable to users. Moreover, it gave poor results on theoretical measures. Weber and Specht [1997] stated that "printed textbooks are not suitable for being transformed to hypertext pages in electronic textbooks in a one-to-one manner". On the other hand, other authors (e.g. DiMarco et al. [1997]) use English text as the source for generating hypermedia pages that are adapted to patients' profiles, for giving them medical information and advice, with successful results.

2.2.2 Techniques for planning the hypermedia pages

One of the first NLG systems that used a model of the user was described by Wallis and Shortliffe [1985]. They modelled two features of the user: the expertise in the subject, and the preferences regarding the level of detail of the explanations; each of this features were modelled with a single number, and the adaptation

was performed by omitting details from the explanation generated by an expert system if the user wanted it less detailed. Since then, many systems appeared which investigate how to model at least one of the user's features in a more sophisticated way, and how to apply that model for language generation.

Paris [1989] observed that the user's level of expertise does not only affect the details of the generated explanations, but also the kind of information: beginners may require a higher level of detail in basics (e.g. process traces) while experts may want a higher level of detail about technicalities. To incorporate this into the system, Paris distinguished two types of entities in the Knowledge Base: basic concepts and specific artifacts. Other approaches included not only detailed information about the user's knowledge in structures such as semantic networks, but also about the user's beliefs and inferences. Milosavljevic [1998] considered the generation of explanations that take into account the similarities between the concepts that the user already knows and unknown concepts, e.g. to distinguish between an unknown concept and a potential confusor, or to describe a concept in terms of a similar concept. McCoy [1989] stores a model of each user's beliefs, in which it allows the users to have a knowledge model different to that of the system, either with the concepts arranged differently in the semantic network, or with different values of the concepts' attributes. Zukerman and McConachy [1993] describe a system that is able to make inferences about the user's knowledge in order to address erroneous inferences and omit easily inferred information.

We can assume that the generation of a hypermedia page usually starts when a user clicks a link requesting some particular information, such as a request for a description of a museum item [Milosavljevic et al., 1998, Oberlander et al., 1998], or when a user asks for information in a medical system [de Carolis et al., 1998, DiMarco et al., 1997]. An equivalent situation happens in real museums, with a personal assistant, when the user is physically standing in front of something about which the system has some information available [Not and Zancanaro, 1998].

The first thing to do is to locate the requested item in the Knowledge Base, and to find a plan in the Plan Library that can be applied to that item. The following are context features that are usually considered when generating the descriptions:

User knowledge has to be taken into account for several reasons. On the one hand, for some information servers, such as museums assistants, it is desirable that the system does not repeat information to the user, as it can become tedious for a visitor to listen to the same information again and again. In this case, if the users ask for information about a particular item, and next ask for further information, they expect a new and more detailed description.

Secondly, user knowledge can be used to describe a concept by comparing it with other items that the user already knows. This was applied both to virtual museums [Milosavljevic et al., 1998, Oberlander et al., 1998] and to electronic encyclopaedias [Milosavljevic, 1998]. For example, the definition in (1), adapted from the entry in the Columbia Encyclopedia, assumes that the user knows what is a *domestic tabby* and therefore compares the *European wildcat* with it.

(1) The **European wildcat** is an European wild cat resembling a large domestic tabby with a heavy tail; its fur is brownish to gray, with a pattern of light stripes. It can and does interbreed with domestic cats.

Thirdly, when the item of interest is related to an unknown item, it is necessary to provide an explanation of this new item so the user understands it. For example, the author of (2) assumes that the reader does not know *Diego Velazquez* and therefore adds a small description of him.

(2) *"Las meninas"* was painted by Diego Velazquez, who is an Spanish painter from the seventeenth century.

If the representation formalism is a semantic network, the user's knowledge can be expressed in terms of the nodes of the conceptual network that formalises the system's knowledge: each conceptual node can have a degree of confidence that the user knows that topic.

A similar feature to the user knowledge is the **dialogue history**, or **linguistic context**. In fact, a user can be supposed to know the last things that have been said to him or her, so the dialogue history can be used with the same purposes as the user knowledge. Furthermore, the linguistic context can be used, during language generation, in order to substitute the names of the concepts with pronouns; or to decide whether to topicalise a concept [Geldof, 2000], if it had just been talked about in the previous sentence.

The **extralinguistic context** refers to the place and time where the user is interacting with the adaptive system. A system for adaptive guidance [Geldof, 2000] might use it for relating the generated text with the context, so as to catch the user's attention. A museum assistant can use the user's location to provide information about the objects nearby.

User interests and goals are sometimes difficult to model, and usually vary greatly depending on the kind of application. For example, if we consider a museum assistant [Not and Zancanaro, 1998], the user profile might include features such as whether the user is interested in some particular style or author, or about the total amount of time that the user wants to spend inside: if there is not much time available, the descriptions have to be shorter. It may also be the case that a visitor wants to find a particular item inside the museum: in this case, the system can provide directions, and very short descriptions of the items located near the user. Other applications may be interested in modelling different features.

According to these factors, the relevant information will be extracted from the Knowledge Base, and then organised using a plan so as to produce a coherent structure to the output. There are several possible ways to organise the facts in a text; Milosavljevic [1998] generates the encyclopaedia definitions mainly with successive comparisons; the approach followed by Mellish et al. [1998] in the ILEX system consists in storing, in the Knowledge Base, together with the facts, the rhetorical relations that hold between those facts. For example, one fact may be an *example* of other fact; or it may be an *elaboration* of other if it further elaborates the information contained in the other fact. These rhetorical relations are then used in order to join the sentences that express different facts with cue phrases (e.g. *for example* or *although*) that give coherence to the text. A more detailed example of rhetorical relations can be found in Section 7.2.2.

2.2.3 Techniques for generating the text

The procedure of **concatenating sentence fragments** is more sophisticated than simply outputting the whole fragment as canned text, but cannot be considered proper Natural Language Generation. In this approach, sentences are divided into fragments or chunks, and a network is used to select the fragments that are more appropriate to the user and the dialogue history. This approach was followed by Not et al. [1998]. Figure 2.5 shows the rules for producing a description of the *Spotted Salamander*, depending on two features: `deictic`, which is true if the user is standing in front of the salamander, and `PRO`, which is true if the discourse context was already about that salamander. Table 2.2 shows the sentences generated for all combinations of values of these two features.

A second approach, called **metatext** [Geldof, 1998], consists in generating small descriptions, using NLG, with hyperlinks to pieces of canned text, or to hypermedia pages on the Internet. For example, Geldof [1998]

```
If +PRO
      Output It
Else  if +deictic
            Output What you are seeing is
      Output The Spotted Salamander
      If +deictic output . It
Output gets its name from its many yellow spots on its black or bluish black body
```

Figure 2.5: Rules for generating different outputs depending on the user profile and the context [Not et al., 1998].

Features	Text
+PRO, +/-deictic	*It gets its name from its many yellow spots on its black or bluish black body*
-PRO, +deictic	*What you are seeing is the Spotted Salamander. It gets its name from its many yellow spots on its black or bluish black body*
-PRO, -deictic	*The Spotted Salamander gets its name from its many yellow spots on its black or bluish black body*

Table 2.2: Different texts generated depending on the user profile and the context [Not et al., 1998].

shows sentence (3) as an example of one of the possible outputs of her system. The underlined fragments are hyperlinks pointing to information pages about them, and the sentence justifies why those hyperlinks are important. This method was also used by Carenini et al. [1990].

(3) Several movies are showing today, but you may be particularly interested in Enfants Perdus, featuring Gérard Depardieu (you may check this opinion here). It shows at 8.30 in Arenberg (see map and public transportation).

To construct the small descriptions, Geldof [1998] used templates, consisting in sentences that had empty slots to be filled in with information from the Knowledge Base, such as "you may be particularly interested in _____". When the sentence is produced, the slots are filled, and several sentences are put together, in a process which may imply the addition of conjunctions, the substitution of repeated proper names with definite Noun Phrases or pronouns, and the ellipsis of repeated information. Busemann and Horacek [1998] emphasise that, from the point of view of applications, domain-specific templates are more efficient than general linguistic knowledge.

A different approach, called **selection and repair**, was used in HealthDoc [DiMarco et al., 1997]. This technique consists in that the system contains, for each piece of information, one or several textual fragments that describe it. If there are several fragments, each one is accompanied with a description about the user for which it was intended (depending on characteristics such as age or expertise). Next, once the system has found all the relevant information, two processes are performed on that information:

1. First, the sentences that contain information relevant to the particular user are *selected*. For example, from the following text [DiMarco et al., 1997], which refers to the risk of having a heart attack,

> Patients who have no history of symptomatic cardiac disease generally have a very low risk of perioperative myocardial infarction and less than a 1 percent risk of death from cardiac causes. However, the risks are much higher in those who are older or have cardiovascular disease.

it is possible that the system only selects the second sentence as relevant for a particular user that has a cardiovascular disease.

2. Secondly, once the relevant sentences have been selected, there is a *repair* step that tries to restore coherence in the resulting text. In the previous example, the following things can be repaired:

   - In the absence of the previous sentence, the conjunction *However* does not make sense, so it should be removed.
   - The definite Noun Phrase *the risks* refers to the *symptomatic cardiac disease*, so that it should be added if the sentence is to be understandable.
   - The comparative adjective *higher* needs that the thing to which it is compared be explicitly said, e.g. *very low* or *less than 1 percent.*

   In the end, what could be expected as the system's output is something like (4):

   (4) The risks of perioperative myocardial infarction and death from cardiac causes are higher than the normal very low level in those who are older or who have cardiovascular disease.

This procedure needs that the knowledge base has complex information about sentence structuring, rhetorical relations between sentences, coreference relations between Noun Phrases, etc. However, authoring is not as difficult as could be imagined, given that much of this information can be acquired automatically with linguistic procedures, with a certain degree of confidence.

True NLG use lexicons and language models in order to generate text from an internal representation, such as a semantic network or FOPL. Once we have a *plan* that indicates which information has to be generated and in which order, it is possible to generate a text describing the contents of that semantic network. This is the approach in the systems described by Oberlander et al. [1998], Milosavljevic et al. [1998] and Milosavljevic [1998].

For example, the plan may consist of the fragment of the semantic network that will be explained to the user, where the nodes in the network represent concepts and the relationships represent predicates between the nodes. A mapping function may link the concepts with nouns and adjectives in a language; the relationships with grammatical constructions (e.g. verb phrases or prepositional phrases); and the discourse plan with expressions that express rhetorical relations (e.g. exemplification, contrast, etc.). With this approach, it should be possible to generate fairly coherent text.

NLG technology provides the following advantages [Milosavljevic et al., 1998]:

- **Description of existing data:** the internal representations of existing knowledge, such as semantic networks or predicate-logics axioms can be used to store many different kinds of information, such as descriptions, numerical data (e.g. stock reports or weather reports), the reasoning of an expert system or documentation for a program. Using NLG procedures, it is possible to generate text expressing all that information.

- **Contextual tailoring**: the generated text is not constrained by existing chunks of natural language data; instead, it is generated dynamically, and it is possible to perform small changes to the sentences and the structure of the text depending on the user profile and the context.
- **Up-to-date documentation**: if the semantic network or the knowledge base that stores the information is updated, the generated texts will reflect the new changes automatically.
- **Multilinguality**: The same concepts in a semantic network, or the same predicate identifiers in FOPL can be mapped to language words using different lexicons, so it is possible to generate different languages automatically.

2.3 Summary

Adaptive Hypermedia is a somewhat recent area that combines expertise both from Hypermedia and from User Modelling. The objective of this discipline is to be able to build hypermedia systems that adapt to the characteristics of the user. It has been applied to hypermedia education, recommendation systems, on-line information systems, information retrieval from large repositories, and on-line help systems, with potential applications in other areas.

In order to be able to adapt to a user, the system should create user models, incorporating all the features it needs to provide the most adequate information. This includes static information, such as the date of birth, the language or the geographical location; dynamic information, such as the users' knowledge, expertise, interests and goals; and environmental information, e.g. the software and hardware used.

From a general point of view, there are two possible adaptations in a hypermedia system: **adaptation of contents**, which can be performed by substituting a text fragment with other, hiding or dimming fragments, sorting fragments, changing the media, or generating the text with Natural Language Generation techniques. The second kind of adaptation is the so-called **adaptive navigation support**, consisting in showing to the users links to the documents that will be more relevant for their purposes, or annotating those links.

Concerning the use of Natural Language Generation techniques for generating the output, it is usually necessary to have a Knowledge Base with information (concepts and facts about them), in some internal representation, from which the information will be obtained. When the system receives a user's petition of information, it must plan the answer document using the user query, the user model and the Knowledge Base. Once it has decided which is the information that shall be produced, it must generate the output in natural language. There are several possible ways to produce the output, such as using templates which will be filled in with words from the user model or the knowledge base; using text that is modified on the fly, or using a language model to translate an internal semantic representation into well-formed sentences.

Part II

Off-line processing: Domain Knowledge Acquisition

Introduction to Part II

This part describes the modules that constitute the off-line processing of the proposed architecture. The purpose of these components is to analyse the source texts in order to obtain all the information that will be used later to generate the hypertext pages. The output of the processing performed here will be in form of annotations in the original text, and several tables in a relational database.

In this step, the following kinds of entities will be identified in the texts:

- Unknown domain-specific terms, such as names of new people, animals and plants, locations (cities, countries, buildings, isles, etc.), bodies of water (rivers and lakes), artifacts, etc.
- Temporal expressions, that shall be used to order the events from the original documents, so the user can browse the information of the documents in chronological ordering.
- Scientific names of animals and plants were also recognised, as they constitute a kind of entities whose names can be easily be identified with specific code.

The first two chapters, 3 and 4, contain two literature reviews: about automatic lexical knowledge acquisition, and about the Distributional Semantics hypothesis, and the methods and techniques derived from it. These are all central to this part of the thesis.

Next, Chapter 5 describes the approaches followed for acquiring new words and inferring their meaning, and Chapter 6 describes other approaches that have been built, mainly with regular expressions and other kinds of patterns.

Chapter 3

Lexical Knowledge Acquisition

A **Lexical Knowledge Base** (LKB) is a repository that contains words and information about them. This information may be of many different kinds. For example, if the knowledge base is used for Natural Language Processing (NLP), then it may embrace several areas from the field, such as phonology, morphology, syntax, semantics, pragmatics, and multilingual links. LKBs can increase notably the power and the accuracy of NLP systems, as they contain information, such as morphological variations, selectional restrictions or synonyms, that is crucial for any sophisticated analysis of texts.

As an example, let us consider the entry in Figure 3.1 that represents the singular third-person present-tense form of the verb **to come**. This entry contains information that is useful for morphological analysis, parsing (the sub-categorisation frame), semantic disambiguation, semantic analysis, and machine translation.

There are several reasons why it is desirable to have an automatic way for building lexicons:

- Constructing LKBs requires a large investment of time and human effort, even if there are other kind of resources available, such as dictionaries and grammars.
- For minority languages it is even more difficult to find and build LKBs. For some of them there are no resources available, and grammarians have to encode all the linguistic information by introspection, by analysing how they speak.
- Many applications need domain-specific dictionaries, and we cannot expect a general-purpose dictionary to contain entries for every possible domain-specific. Furthermore, neologisms appear all the time, which makes it impossible to be always up to date.

3.1 Representation of a LKB

LKBs, also called lexicons, can be classified in two groups depending on the way in which they are organised. *Word-based* or *lexical-based* lexicons are structured according to the words they contain; while *concept-based* lexicons are structured around the meaning of the words. For example, a lexicon that contains words, together with all the possible morphological analyses of those words is structured around the words themselves; it does not matter whether two words have the same meaning or not. That lexicon would be word-based. On the other hand, in a concept-based lexicon, concepts are organised according to their semantics. For example, in one such lexicon, the concept *dear* (beloved) will be located closer to *lovely* (beloved) than to

Word: comes
Phonology: /cams/
Morphology: present-tense verb, 3rd person singular, stem=come
Syntax (sub-categorisation frame):
    Specifier: [NP-Subject]
    Complement: [PP(to)]
Semantics:
    $\lambda x.\lambda y.come(y, x)$
    Hyperonyms (subsumes this concept): **{travel}**
    Hyponyms (concepts subsumed): **{approach, emanate, come near}**
    Antonyms: **{go}**
Selectional restrictions:
    Subject ϵ **animated-being**
    Complement ϵ **location (destination)**
Multilingual:
    Spanish: viene
    French: viens
    Catalan: ve
    Basque: etortzen da
    Asturianu: vien
    Galician: vén
    Portuguese: vem

Figure 3.1: Example of entry in a Lexical Knowledge Base, corresponding to the word *comes*

dear (expensive). That is, words sharing their meaning will be related in the lexicon, and words with a different meaning will be far apart, even if they have the same lexical form.

Lexicons that contain, for each entry, a well-defined set of characteristics, can be stored in a relational database. A few examples of these lexicons are the following: Atserias et al. [1998] describe the MACO+ morphological analyser that stores, in a LKB, all possible inflected forms of words with their parts-of-speech and morphological features; and the syntactic analyser *grok*, from the university of Edinburgh, that stores, for every word, how it combines with other words to form complex constituents (its sub-categorisation frames). Machine-Readable Dictionaries (MRDs) and thesauri can also be easily stored in relational databases.

However, when the complexity of the information we need to store increases, we have to switch to a more robust knowledge representation formalism. In some cases, not all the information is stored in the lexicon, because some of the data can be inferred from the LKB and it is not necessary to codify it explicitly. For example, some information might be encoded as First Order Predicate Logics (FOPL) axioms, and the lexicon manager would have to decide whether some additional information can be derived from the axioms or not. If a system needs to make inferences, then **Object-oriented data bases** and **Knowledge Representation Systems** (such as frame-based Systems) are more suitable frameworks.

3.2 Conceptual ontologies

Concept-based LKBs are usually organised as semantic networks, nets of nodes and arcs where nodes represent concepts and arcs represent relationships (see Figure 3.2a). If two concepts are semantically very related (e.g. if one represents a specialisation or a generalisation of the other concept) there will be arcs joining them, while if they are completely unrelated then the paths from one to the other in the network will be

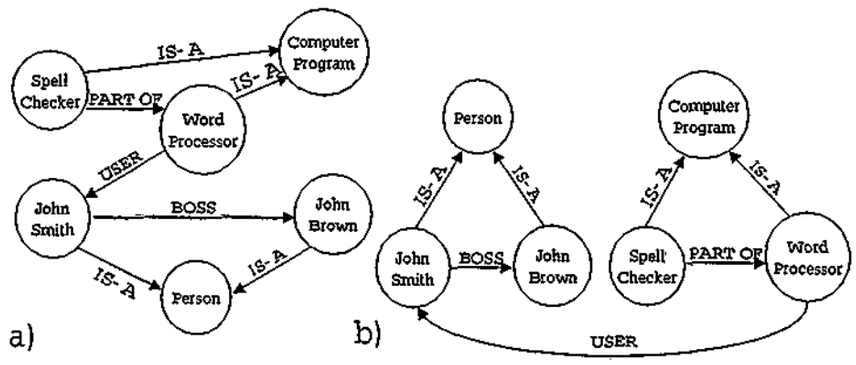


Figure 3.2: (a) Semantic network of concepts and relations amongst them. (b) The same semantic network, redrawn to make evident that the IS_A relationship establishes a taxonomy.

long. It is common for these LKBs to be called *ontologies*. There have been many definitions of ontologies in the Knowledge Representation literature, and the following is one of the most accepted ones [Gruber, 1993]:

> An ontology is an explicit specification of a conceptualisation. The term is borrowed from philosophy, where an ontology is a systematic account of Existence. For knowledge-based systems, what "exists" is exactly that which can be represented. When the knowledge of a domain is represented in a declarative formalism, the set of objects that can be represented is called the universe of discourse. This set of objects, and the describable relationships among them, are reflected in the representational vocabulary with which a knowledge-based program represents knowledge. Thus, we can describe the ontology of a program by defining a set of representational terms. In such an ontology, definitions associate the names of entities in the universe of discourse (e.g., classes, relations, functions, or other objects) with human-readable text describing what the names are meant to denote, and formal axioms that constraint the interpretation and well-formed use of these terms.

Because a semantic network can contain the objects of interest and the relations amongst them, an ontology can be represented using this formalism. The term *ontology* is usually used in the particular case in which (at least) one of the relationships establishes a taxonomy. The typical example of ontologies are those in which the taxonomy is created with the relation of *generalisation*, i.e. when the conceptualisation is structured from the more general to the more specific concepts (see Figure 3.2b).

3.2.1 Information in an Ontological LKB

When an concept-based LKB is structured as an ontology, it can include some or all of the following information:

- **Words**: the lexical form of the words in human languages.

- **Concepts and Instances:** There is disagreement about the interpretation of instances and concepts, but one of the most widely accepted considers concepts as the sets of their instances [Montague, 1974]. With this interpretation, a concept represents a set of things of interest that have something in common, and each of the elements in the set is an instance. For example, *human* is a concept that can denote the set of every instance from the *Homo sapiens* species.

  Instances are single examples of concepts. For example, *John Smith* may be an instance of the classes *person*, *sneaker*, *bookshop*, etc.

- **Relations:** There are many different kinds of relationships that can be represented in a semantic LKB. The basic relationship in ontologies is the *specialisation*, that produces the taxonomic structure by linking the general concepts with the more specific ones. There are many possible kinds of relationships that can be encoded in an ontology, such as antonymy, aggregation, location, agent, etc.

  In a semantic network, different relationships are encoded with different classes of pointers between concepts and instances. On the other hand, ontologies that are based in FOPL represent relationships as predicates and functions. A predicate has one or more arguments, and evaluates to a truth value. For example, the predicates IS_A or IS_A_KIND_OF can relate a concept with all its sub-concepts, forming a taxonomy. A function, on the other hand, takes concepts or instances as arguments, and returns other concept or instance as a result. For instance, the function FATHER can take an instance of an *animal* as argument and return the instance which is its father.

Note that a single lexical word may refer to several concepts or instances. For instance, the concept *horse* can be used to refer to:

1. every animal in the *Equidae* family, species *Equus caballus*, herbivorous, usually domesticated and used as a beast of burden and for riding.
2. every frame –usually with legs– used for supporting something (e.g. sawhorse, gymnastic horse, etc.)
3. every chess piece (usually resembling a horse) that always moves one step straight and one step athwart.
4. every sample of heroin.

On the other hand, several words can refer to the same concept, e.g. *horse*, *junk* and *heroin*; although the use of one or other word may provide additional information such as the speakers social level or geographical origin, or their attitude toward it.

For this reason, a conceptual ontology has to include a mapping from the lexical forms of words into the concepts that they may represent.

3.2.2 Applications

Conceptual ontologies have been applied to solve problems in different fields, from electronic commerce to corporative knowledge management. In Natural Language Processing, they are a useful tool as a source of linguistic knowledge for many kinds of applications, such as the following:

- **Information Extraction (IE):** IE consists in locating and extracting, from a document, the information that is relevant for a user. For example, a user might want to know about all the joint venture operations that appear in an issue of a financial newspaper. An IE system can be asked to find them,

and to return, for each operation, the date and location at which it happened, and the companies and amounts of money involved.

Much of the early research on IE was done in the Message Understanding Conferences (MUC) [MUC6, 1995, MUC7, 1998], competitions at which different systems were evaluated. It was soon observed that semantic dictionaries can help in many ways. For example, they could be used to find specialisations of the concepts of interest (e.g. if looking for employees, a LKB might contain the information that a *clerk* is an employee). Semantic categories can also be included in the preconditions of extraction rules, so these are more general [Soderland, 1999] (e.g. a rule may state that if a verb is more specific than *to attack*, then its subject is probably of interest to the user).

- **Information Retrieval (IR):** IR is the academic discipline that studies procedures to locate relevant information in huge repositories of data. Given a query provided by some user, its aim is to find the documents that are more relevant for him or her. LKBs can be very useful tools for IR [Voorhees, 1993]. They can be used for recognising synonyms, morphological variations or specialisations of the query words; and for identifying the genre of a text [Baeza-Yates and Ribeiro-Neto, 1999]. The benefits of multilingual LKBs for multilingual IR are also evident [Hull and Grefenstette, 1996].

- **Question-Answering (QA) and Knowledge Retrieval:** The aim of QA systems [Voorhees, 2000] is to find the answer to a question inside a collection of texts. A typical procedure consists in calculating some semantic similarity metric between the question and the sentences in the texts, and returning the sentence that ranked first. LKBs can be applied to Question Answering systems in order to calculate this similarity between sentences [Clark et al., 1999, Cohen et al., 1999, Alfonseca et al., 2001].

- **Word Sense Disambiguation (WSD):** WSD is the task that consists in finding the sense with which a word is being used in some context. For example, the meaning of *bank* in sentence (5a) is the shore of the river; while in sentence (5b) it means a financial institution.

  (5) a. Mary walked along the bank of the river.
      b. HerborBank is the richest bank in the city.

  Concept-based lexicons can provide, for each lexical word, an enumeration of all the concepts that can be represented with that word. Therefore, WSD can be defined as a classification problem, where all the possible classes for each word are given by the LKB.

  Because LKBs also provide additional information, such as which other concepts are semantically related with each one of the possibilities, that information can be used for choosing the sense such that its related concepts in the LKB are related to the context words [Yarowsky, 1992, Agirre and Rigau, 1996, Wilks, 2001]. In the example above, in (5a) *river* indicates that *shore* is the right sense; while *richest* in (5b) supports the election of *financial institution* as the right sense.

- **Machine Translation (MT):** A multilingual semantic dictionary contains relations between senses of words in different languages. For instance, a dictionary with English and Spanish words can have a link from *bank* (as a lexicalisation of the concept *shore*), to the Spanish word *orilla*; and a link from *bank* (as a lexicalisation of *financial institution*) to the Spanish word *banco*. These LKBs, combined with WSD techniques, are very useful resources for Machine Translation [Agirre et al., 2000b].

Additionally, conceptual ontologies have been applied to other fields such as automatic text summarisation [Mani, 2001] and for dialogue interfaces [Rosé, 2000].

Relation	Symbol	Description and examples
hyperonymy	@→	Links a synset to its immediate generalisations. **artisan @→ worker**
hyponymy	~→	Links a synset with its immediate specialisations. **artisan ~→ {book-binder, clock-smith, ...}**
meronymy	#→	IS_PART_OF relationship. It includes: * set membership: **parent #m → family** * component-whole: **wheel #p → wheeled-vehicle** * constituent stuff: **water #s → teardrop**
holonymy	%→	HAS_THE_PART relationship (opposite to meronymy) **teardrop %s → water**
antonymy	!→	Links a synset to the one with the opposite meaning. **descent !→ ascent**

Table 3.1: Noun relationships in WordNet. Each relation is represented with a symbol.

3.3 Existing conceptual ontologies

This section contains an overview of some LKBs that have been applied to NLP. The first one, WordNet, is described in more detail as it is the one that will be used in the rest of this work. Next, a small overview is provided about other lexical resources.

3.3.1 WordNet

WordNet [Miller, 1995] is a semantic network of concepts that was built on psycho-linguistic grounds. It is a concept-based general-purpose ontology.

In WordNet, words are grouped in sets of synonyms that represent the same concept, called *synonym sets* or *synsets*. Each synset contains, apart from the words, a brief definition. The version used in this work, 1.7, has 109,376 different synsets that cover nouns, verbs, adjectives an adverbs. Of all these, we shall focus on the hierarchy of nouns.

WordNet 1.7 contains a total of 74,487 noun synsets. Table 3.1 describes the relationships that we can find between nouns. Two opposite relationships organise the concepts in a taxonomy: the relation of generalisation, which is called *hyperonymy* or *hypernymy*, and the opposite relation, specification, which is called *hyponymy*. If a synset s_1 is more general than a synset s_2, then s_1 is on top of s_2 in the hierarchy, and we say that s_1 is a hyperonym of s_2 or that s_2 is a hyponym of s_1.

The PART_OF relationship, of which several subtypes are considered, is called *meronymy*, and its opposite *holonymy*. Finally, antonyms are also marked with other relationship. Each of the relations is identified with a different symbol.

We can define the hierarchy of nouns in *WordNet* as a semantic network $\mathcal{W} = (\mathcal{L}, \mathcal{S}, f_{\mathcal{L}}, h_{\mathcal{S}}, \mathcal{R})$ where

- $\mathcal{L}$ is the set of lexical entries (words).
- $\mathcal{S}$ is the set of synsets.
- $f_{\mathcal{L}} : \mathcal{L} \rightarrow \mathcal{S}^+$ is a function that links the lexical entries with the synsets that contain them.
- $h_{\mathcal{S}} : \mathcal{S} \rightarrow \mathcal{S}^*$, *hyperonymy*, arranges the concepts and instances in a hierarchy.
- $\mathcal{R}$ is the set of other relationships.

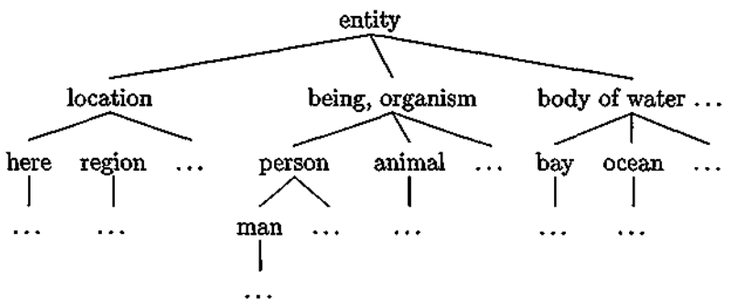


Figure 3.3: A small portion of the WordNet taxonomy that is rooted on the synset *entity*.

Hyperonymy has been given special relevance in the definition because it is generally used to structure the synsets as a taxonomy, although other relations, such as *meronymy*, can also be considered taxonomic.

Nouns are structured in several separate taxonomies, i.e. there are several synsets which have no hyperonyms. Also, the hyperonymy relationship does not arrange synsets as a tree, because a single synset may have several hyperonyms. Figure 3.3 displays a portion of the upper part of the taxonomy that is rooted at the synset *entity*, and which contains most of the concrete nouns such as locations, life forms or artifacts. Other sub-taxonomies contain concepts that are *psychological features*, *actions*, *events*, *abstractions*, etc.

3.3.2 Other existing lexical resources

This section provides an overview of other lexical resources available.

Comlex

COMLEX [Macleod and Grishman, 1994] is one of the first machine-readable lexicons built for Computational Linguistics. The 35,000 entries it contains for English headwords have the following information:

1. Inflected forms of the words (plural of nouns, inflection of verbs).
2. Sub-categorisation frames for verbs.
3. Other features, such as which nouns are mass nouns, or units; control information in verb frames; etc.

COMLEX is a word-oriented lexicon, i.e., it just contains English words together with morphological and syntactic information, but it does not include information about the concepts they represent. Synonym words are treated as different words, while polysemy is not addressed.

CoreLex

CORELEX [Buitelaar, 1998] was an attempt to change WordNet into a Generative Lexicon [Pustejovsky, 1995]. The concepts from WordNet were grouped in 45 basic types and 326 combinations of them, using automatic algorithms. CORELEX is distributed in the form of a list of WordNet concepts together with the group to which each one belongs.

Because every one of those groups, by definition, has some properties associated, such as argument structure or semantic interpretation, the lexicon conveys much information that WordNet lacks. However,

to the author's knowledge, there are but a few applications of this LKB (one of the few examples is described by Bouillon [1999]).

Acquilex

The ACQUILEX projects, funded by the European Union, centred on automatic acquisition of lexical knowledge from MRDs. They lasted from 1991 to 1995, and the result was a collection of tools for storing lexical entries, parsing dictionary entries and automatic indexing for several European languages. One of the products of these projects is the LKB system [Copestake, 1992] [Copestake et al., 2000] for parsing and semantic interpretation.

Cyc

Cyc [Lenat, 1995] is a general-purpose lexicon which started in 1984 at MCC in Austin, under the direction of Doug Lenat. The lexicon is still being built, after investing on it more than one person-century of effort. Internally, it is subdivided into micro-theories each one of which contains knowledge (concepts, instances and axioms) about a particular domain. The Upper Ontology contains general concepts that are likely to be useful in every domain, such as the top nodes in the hierarchies. It is a commercial product, but the upper ontology can be freely distributed.

The purpose of Cyc goes well beyond that of other semantic lexicons such as WordNet. WordNet basically contains the hierarchy of concepts and a few relationships amongst them, whilst Cyc includes a whole range of relationships and predicates to be able to represent most natural language utterances in an extension of FOPL called CycL. We can summarise the way information is stored in Cyc in the following points:

- Words are stored in the Cyc Lexicon, which contains around 32,000 different words.
- Each lexical word can have different denotations or concept meanings, which are distinguished by an integer identifier.
- Concepts and instances are organised in a hierarchy.
- Each concept also has a sub-categorisation frame specifying its complements and the thematic role of each one. Instead of having a fixed number of thematic roles, such as LOCATION, AGENT, PURPOSE, etc. the thematic role can be any hyponym of the concept **ActorSlot**, thence it is an open set.
- Global rules and axioms written in FOPL infer new information as the system learns more knowledge.

Table 3.2 shows the number of concepts contained in some of the lexicons discussed.

Other lexical resources

There are other commonly-used lexical resources, such as the UMLS (Unified Medical Language System) project, supported by the National Library of Medicine [UMLS, 1998]. It is a long-term project that includes more than 800,000 medical terms organised as a semantic network. Its aim is to facilitate the retrieval and integration of machine-readable biomedical information sources.

Part of speech	WordNet 1.6	WordNet 1.7	Cyc	Comlex
Verbs	12,000	12,754	4,000	8,000
Nouns	60,000	74,487	17,700	21,000
Adjectives	16,500	18,523	4,000	6,000
Adverbs	3,000	3,612	2,000	
Multiword Phrases			6,300	
Total	91,500	109,376	34,000	35,000

Table 3.2: Comparison of the sizes of some LKBs. The numbers are approximate, except in the case of WordNet 1.7.

The MikroKosmos project [Nirenburg et al., 1996] is developed jointly by researchers at New Mexico State University, Carnegie Mellon University and various U.S. government agencies. It aims at defining a methodology for representing the meaning of texts in an interlingual format called Text Meaning Representation that is language-neutral.

Other multilingual lexicon is SIMPLE [SIMPLE, 2002]. SIMPLE is an European project to add semantic information to the lexica built for 12 European languages by the PAROLE consortium. PAROLE +SIMPLE lexicons contain morphological, syntactic and semantic information, organised according to a common model and to common linguistic specifications.

3.4 Acquisition of Conceptual Ontologies

An important step in the construction of complex software systems is the encoding of knowledge in such a way that it can be provided to and processed by a software component. This involves many kinds of knowledge, such as decision trees or rules for classification systems or for expert systems (e.g. medical diagnosis programs), spatial and temporal knowledge for scheduling algorithms, or linguistic knowledge for NLP applications. Traditionally, knowledge was acquired by the so-called *knowledge engineers* either from text books or by consulting an expert in the field. The task of gathering this information is called *Knowledge Acquisition* (KA). The sub-task of gathering information from the expert is called *Knowledge Elicitation* (KE). What differentiates knowledge-based system development from conventional system development is the emphasis on in-depth understanding and formalisation of the relations between the conceptual structures underlying expert performance and the computational structures capable of emulating that performance.

It is widely agreed [Clark et al., 1999] [Webb et al., 1999] that KA is the bottleneck of many kinds of knowledge-based systems. In the same way, the acquisition of lexical resources is a bottleneck for building NLP applications. There are many recent research projects addressing the topic of automatically extracting taxonomies from texts. It is possible to automatise the acquisition of many kinds of information that can be stored in a LKB: selectional restrictions [Resnik, 1993, Faure and Nédellec, 1998], proper nouns [Mikheev et al., 1998], collocations [Church et al., 1991], syntactic rules [Hockenmaier et al., 2000, Briscoe and Carroll, 1997, Xia, 1999], multilingual links [Dagan et al., 1991], new word senses [Basili et al., 1998] or non-taxonomic relationships [Maedche and Staab, 2001].

There is also a need for porting existing LKBs to other human languages apart from English, and also to extend English resources with domain-specific information and updated data. One of the largest research projects for building conceptual ontologies is the European project called EuroWordNet [Vossen, 1998], which is in the process of building wordnets for several European languages (Dutch, Italian, Spanish,

German, French, Czech and Estonian). There are groups working in porting WordNet to languages such as Oriya [Mohanty et al., 2002], Tamil [Poongulhali et al., 2002] and Hindi [Chakrabarti et al., 2002], amongst others. However, building such resources is a labour-intensive task that requires the participation of expert lexicographers that are also familiar with computational ontologies.

The need of wordnets in several languages was answered with the development of algorithms that, using the English WordNet and machine-readable dictionaries (mono-lingual [Rigau, 1998] or bilingual [Daudé et al., 2000]), automatically port it to other languages. Daudé et al. [2000] describes a procedure to define mappings between the English version of WordNet and Catalan; and that same approach proved also valid for a language as distant to Catalan as Korean [Lee et al., 2000].

However, there is still much work to do, as it is still difficult to extend automatically an existing WordNet with new domain-specific words, if a domain dictionary is not available. This section contains a brief literature review on the field on extracting information into ontologies.

3.4.1 A classification of Ontology Learning algorithms

Ontology Learning algorithms can learn many different features of ontologies, but this review centres on the systems to build automatically the taxonomic structure based on the hyperonymy/hyponymy relationship. We can classify these approaches according to the following criteria:

1. Automaticity.
2. Input data.
3. Approach taken.
4. Operations performed to the ontology.
5. Degree of supervision.
6. Machine Learning method used.

Automaticity

The ontologies that have been developed by hand, such as WordNet, Cyc or Comlex, were created by lexicographers using *introspection*. The creators of the LKB transferred to the ontology *their own knowledge* about the language and the words. This is the most reliable way of making sure that the ontology is correct, but it has several drawbacks:

- Different lexicographers may differ in their opinion.
- Although this process is usually machine-aided, with tools to navigate ontologies and check their consistency, it is still highly resource-consuming, as the development time of ontologies built with this method shows.

It is possible to build wordnets for languages for which there are enough resources, such as Spanish, English, German or Italian. However, the investment needed to build an ontology for minority languages and languages that are spoken in poor regions is excessive. Therefore, it is necessary to automatise as much as possible the task of Lexical Knowledge Acquisition (LKA) in order to reduce development costs. According to automaticity, we can classify systems in two types:

- *Automatic*: the output produced by the systems, although it is expected to contain a certain amount of mistakes, does not necessarily need user intervention [Rigau, 1998, Hastings, 1994, Hahn and Schnattinger, 1998].

- *Semi-automatic*: these systems require either that a judge-user validates the output *suggested* [Assadi, 1998, Kietz et al., 2000] or that the user completes this output, as in ASIUM [Faure and Nédellec, 1998].

Input data

MRDs have been used for some time to learn ontologies, due to their availability in electronic form, and to the fact that dictionary definitions are structured enough as to be able to discover hyperonymy relationships from them. In dictionaries, a concept is usually defined with a more general term, and by specifying the particular characteristics that differentiate it from its hyperonym. For example, the following definition from the Merriam-Webster dictionary provides the information that *herb* is a generalisation of *lily*.

> **Lily**: any of a genus (Lilium of the family Liliaceae, the lily family) of erect perennial leafy-stemmed bulbous *herbs* that are native to the northern hemisphere and are widely cultivated for their showy flower

Using automatic methods, the hyperonym can be found in the definition; it has to be semantically disambiguated, and then the hyperonymy relation between these two concepts can be induced. Therefore, a whole ontology can be constructed from a dictionary [Wilks et al., 1996, Rigau, 1998].

Learning from MRDs has obtained the best results when combined with existing ontologies. For example, Rigau [1998] generated wordnets for Spanish and Catalan using the original WordNet, bilingual dictionaries and mono-lingual dictionaries. He reports that the more dictionaries he uses, the better the reliability of the generated results. The use of the original English WordNet is very important because dictionary definitions alone usually produce shallow ontologies, they are sometimes inconsistent, and can contain circular definitions (when a concept is defined in terms of other and vice versa).

Other systems learn from unrestricted text. Some of them look for language patterns in the texts that provide information about which is the hyperonym of an unknown concept [Hearst, 1992, Hastings, 1994, Hahn and Schnattinger, 1998]. Others use co-occurrence information or selectional restrictions to cluster the words in function of the similarity of the contexts in which they are used, such as ASIUM [Faure and Nédellec, 1998] or the research by Lee [1997].

Finally, good results can be obtained with a mixed approach. Kietz et al. [2000] reports about a system that first learns hyperonymy relationships from dictionary definitions, and next analyses free text from a corporate intranet to find more relationships.

Approach

In the **prescriptive approach**, a world-model is generated before completing it with lexical entries. First, concepts are identified in the real world; they are then structured in an ontology; and at the end these are associated with the words that are used to express them. In this framework, the representation of the concepts is independent of their lexical form, and words are linked to concepts through a lexical mapping

$$\mathcal{L}(w) = \{c : w \text{ lexicalises } c\}$$

This was the approach taken for building Cyc [Lenat, 1995] and WordNet [Miller, 1995]. The approach is very useful for building multilingual ontologies, because the conceptual structure is independent of the language. In EuroWordNet [Vossen, 1998] there is an inter-lingual index (ILI) that contains the concept structure, organised as an ontology, and which is independent of the lexicalisations. For each of the languages, they are structured as independent ontologies, and there are multilingual links between the language-dependent synsets and the synsets in the ILI. A concept from the ILI might have lexicalisations in all of the languages or only in some of them; but every concept that is lexicalised in at least one language appears in the ILI.

On the other hand, in the **descriptive approach**, concepts are inferred from words. Most of the clustering algorithms apply this approach. Lee [1997] clustered words according to the contexts in which they appear; and the system ASIUM [Faure and Nédellec, 1998] clusters nouns and verbs to form an ontology according to their selectional preferences: verbs that can accept the same types of nouns are considered similar, and they are clustered together. One of the main drawbacks of this approach is that it is very difficult to discover whether a word is behaving polysemously or not.

Other works follow a combination of both approaches. These systems can start with the definition of the top-level concepts, following a prescriptive approach, and then attach the lexical data to it, possibly giving it structure with a descriptive approach. For example, Resnik [1993] and Li and Abe [1997] combine WordNet and a mono-lingual corpus with distributional statistics to obtain selectional restrictions.

Operations performed

There are, at least, three distinct operations that can be performed to ontologies: building, merging and refining.

Ontology Building consists in constructing an ontology from scratch. This includes the manual method of introspection used for the construction of WordNet or Cyc, and some automatic methods, such as *clustering*, that organise a set of initially unstructured words or concepts into an ontology [Lee, 1997, Faure and Nédellec, 1998], or analysis of MRDs [Wilks et al., 1996, Rigau, 1998].

Ontology Merging consists in putting together the information contained in two or more ontologies. For this task, the nodes that refer to the same concept will be merged; and some relations have to be found amongst the remaining nodes from the different ontologies [Roventini et al., 2002, Magnini and Speranza, 2002].

Ontology Refinement (OR) [Maedche and Staab, 2001], is defined as the adaptation of an ontology to a specific domain or to some user's requirements, without altering its overall structure. For this second task, it is assumed the existence of an initial ontology already built, and the system has to remove irrelevant terms and to add domain-specific or user-specific concepts. This is the task performed by Kietz et al. [2000] and Navigli and Velardi [2002].

Degree of supervision

Machine-Learning techniques can be classified in two types: supervised and unsupervised.

A **supervised** learner needs, in order to be trained, that the user feeds it with a series of inputs and the desired output for each input. The aim of the learner is to be able to produce the correct output for each input in the training data, but generalising, so as to be able to handle previously unseen data. For example, Esposito et al. [2000] uses Inductive Logic Programming to learn rules to classify unknown entities from texts in some pre-defined classes.

Systems that use hand-crafted rules, scripts or heuristics, can also be considered supervised [Hastings, 1994, Hahn and Schnattinger, 1998, Rigau, 1998, Navigli and Velardi, 2002].

An **unsupervised** algorithm builds, from the input, representations that can be used for reasoning. Examples of unsupervised learners are ASIUM [Faure and Nédellec, 1998] and the system by Lee [1997], both of which cluster concepts according to how they are used in texts.

Other systems, such as the one described by Kietz et al. [2000], combine hand-crafted rules and unsupervised methods in the same system.

Learning method

The procedures that learn how to structure ontologies can also be classified in function of the learning paradigm used.

One of the most popular methods is clustering, with two possible variants: bottom-up and top-down. The bottom-up approach [Faure and Nédellec, 1998, Lee, 1997, Assadi, 1998] is taken when all the concepts are considered initially structured as many one-element sets, and the algorithm proceeds merging the most similar sets until they are all merged; the order in which they have been merged determines the hierarchy. On the other hand, the top-down approach is taken when all concepts are considered initially members of a set that contains them all, and which represents the root of the taxonomy. This set is then subsequently split until all the concepts are in one-element sets. Again, the order in which the splitting happens determines the final taxonomy. It must be noted that the generated taxonomies, using these methods, are always trees; it is not possible for a concept to have two different hyperonyms.

Bottom-up clustering is a fairly popular unsupervised method. However, systems based on clustering have a drawback: the concept which represents the result of merging two other concepts does not have a lexical form, so it is necessary that a human annotator provides a name for it; and it might be the case that the merged concept has no counterpart in the language [Faure and Nédellec, 1998]. For example, if *motorbike* and *truck* appear in the same kinds of contexts, they can be merged by the algorithm, but the resulting generalisation of both has to be given a name by the user, e.g. *motorised_vehicle*. In this respect, systems that learn hyponymy relations from MRDs or free texts are better because they automatically discover which words represent general concepts and which ones represent their specialisations.

Classification algorithms for extending existing ontologies [Hastings, 1994, Hahn and Schnattinger, 1998] take an existing ontology and a new unknown concept u and decide which concept, in the ontology, is the one most similar to u. Next, they create a hyperonymy relation between the concepts.

Frequency-based methods discover which are the relevant terms and the relations between them by performing a frequency analysis of different aspects of the texts. For example, they can find relations between nouns and verbs by looking for the most frequent subject-verb and verb-object relationships [Grefenstette, 1993], and they can decide that a concept will be introduced in a domain-specific ontology if its frequency in the domain-specific texts is greater than its frequency in general-purpose texts, e.g. in big general corpora [Kietz et al., 2000].

Other simple method consists in learning textual or syntactic patterns that in some cases may express a hyperonymy relation, such as appositive constructions [Hearst, 1992].

work	aut.	source	approach
[Rigau, 1998]	yes	M.R.D.	both
[Grefenstette, 1993]	yes	parsed text	descriptive
[Faure and Nédellec, 1998]	semi	parsed text	descriptive
[Hearst, 1992]	*semi	free text	prescriptive
[Hastings, 1994]	yes	parsed text	prescriptive
[Hahn and Schnattinger, 1998]	yes	parsed text	prescriptive
[Lee, 1997]	yes	free text	descriptive
[Kietz et al., 2000]	*semi	M.R.D./text	both
[Esposito et al., 2000]	yes	parsed text	prescriptive
[Assadi, 1998]	semi	free text	descriptive
[Navigli and Velardi, 2002]	semi	free text	prescriptive

Table 3.3: Comparison of algorithms for creating or extending ontologies. The second column shows whether they are automatic or semi-automatic (when they require a human judge to validate their decisions or to name the generated concepts). The approaches marked as **semi* can be considered fully automatic, but due to the large amount of errors in the results, the authors recommend that a judge validates them. The third column shows the data on which they feed, and the fourth column shows the approach taken.

work	op.	sup.	method
[Rigau, 1998]	build	yes	hand-crafted
[Grefenstette, 1993]	build	no	frequencies
[Faure and Nédellec, 1998]	build	no	clustering
[Hearst, 1992]	refine	no	patterns
[Hastings, 1994]	refine	yes	classification
[Hahn and Schnattinger, 1998]	refine	yes	classification
[Lee, 1997]	build	no	clustering
[Kietz et al., 2000]	refine	both	hand-crafted/patterns/ frequencies
[Esposito et al., 2000]	IE	yes	ILP
[Assadi, 1998]	build	no	clustering
[Navigli and Velardi, 2002]	refine	yes	heuristics

Table 3.4: Comparison of algorithms for creating or extending ontologies. The second column shows whether they build an ontology from scratch or refine an existing one; the third column shows whether the learning is supervised, and the fourth column shows the methods used.

Discussion

There have been many approaches to automatic LKB building and refining, from different perspectives and using different methods. In the particular case of discovering hyperonymy relations for constructing or extending taxonomies, there is also a great variety of approaches, of which a representative sample has been chosen for this section. Tables 3.3 and 3.4 display a summary of the systems described here.

The algorithms for extending existing ontologies will be studied in more detail in the following section.

3.4.2 Ontology Refinement

As already stated before, OR consists in tuning an ontology to the special needs of a user, or to a specific domain of knowledge. One of the first attempts to extend WordNet with domain-specific information was reported by O'Sullivan et al. [1995], who added new synsets about word processors and software applications.

The work was done by hand by domain experts, but it showed that WordNet is easily extensible with new synsets.

There are only a few publications on automatic OR, which can be classified in the following two groups, according to the kind of output they produce:

- **Deterministic** systems provide, for each unknown concept, a list of hyperonyms all of which are supposed to be correct. Hearst [1992] and Kietz et al. [2000] both use hand-crafted regular-expression patterns in English which usually convey the hyperonymy relation.
- **Non-deterministic** systems provide a set of likely candidates, some of which are correct, either with probabilities or assuming a uniform distribution. Hastings [1994] and Hahn and Schnattinger [1998] used hand-crafted rules that were used to shrink the hypothesis space as more evidence was provided from the texts.

The following subsections describe the main families of OR systems.

Word patterns

The system described by Hearst [1992, 1998] obtains regular expression patterns from free texts by looking at pairs of (hyperonym, hyponym) that co-occur in the same sentence, and then uses them to learn new hyperonymy relations. For example, the following sentence [Hearst, 1998]

> ...works by such authors as Herrick, Goldsmith and Shakespeare...

can be used to find that the pattern `such NPs as {NP,}* NP` usually states a hyperonymy relation. However, she notes that these extracted relations contain a high degree of noise, either because the extracted relation is far too general (e.g. `hyperonym(exercise, thing)`); because they are subjective opinions with little interest (e.g. `hyperonym(Gaslight, classic)`, referring to the film *Gaslight*); or because they result from parsing errors.

Kietz et al. [2000] applied a combination of hand-coded text patterns and an analysis of MRD definitions for extending GermaNet (a German equivalent of WordNet) with concepts from a corporate intranet. He quantified the error rate of the hand-coded patterns in 32%. Also, this procedure cannot find the hyperonym for all the concepts because many of them do not appear in any of the pre-defined patterns. Therefore, all the concepts had to be ultimately revised and placed in the hierarchy by the user.

Navigli and Velardi [2002] consider string inclusion as an indicator of a hyperonymy relation. So, for example, *car ferry service* would be a hyponym of *ferry service*, which would be a hyponym of *service*. They automatically acquire domain-specific multi-word terminology, and next apply a WSD procedure to decide which of the WordNet synsets is the appropriate hyperonym of the new terms. In the example above, the system has to decide which of the synsets containing the word *service* is the one that is really the hyperonym of *ferry service*.

Classification

Hastings [1994] describes a framework for learning nouns and verbs built on top of the LINK NLP system [Lytinen, 1991]. In his system, called Camille, he has concept ontologies for nouns and verbs about the terrorist domain, and the verbs are annotated with selectional preferences, e.g. the object of *arson* is known

Approach	Method	Ontology	Corpus
Hearst [1992]	Det.	WordNet	Grolier's Academic American Encyclopedia
Kietz et al. [2000]	Det.	GermaNet	corporate intranet
[Navigli and Velardi, 2002]	Det.	WordNet	Tourism domain
Hastings [1994]	Prob.	LINK hierarchy [Lytinen, 1991]	newswire articles
Hahn and Schnattinger [1998]	Prob.	KL-ONE Terminological Knowledge Base [Woods and Schmolze, 1992]	I.T. magazines

Table 3.5: Comparison of different approaches to Ontology Refinement. The second column indicates whether the method is probabilistic or deterministic; the third column shows the ontology that was extended, and the fourth column is the corpus that was used to find new concepts.

to be a *building*. Therefore, if an unknown word was found being the direct object of *arson*, the system can classify it as a building.

Initially, every concept in the ontology is a possible hyperonym candidate for an unknown word u. For each appearance of u in a sentence, the verbs for which it is subject or object reduce the possibilities about the class to which it belongs, using their selectional restrictions.

A second version of the system, Camille 2, uses script structures, which specify typical sequences of events, to analyse newswire articles about terrorism. For example, a news about an investigation is said to include the steps of *investigation*, *questioning*, *charging* and *trial-script*. These proved to increase the overall results, but it might increase the difficulty of porting the system across domains.

A very similar approach was taken by Hahn and Schnattinger [1998]. The pre-requisite is an ontology about electronic devices, and a list of constraints such as verbal selectional restrictions. Every time the system finds an unknown word, the restrictions reduce the hypothesis space of possible hyperonyms. For example, a rule can state that if something is in genitive case (either Saxon genitive or using by the particle *of*), then it has the role *possessor*; and other rules state that some kind of entities cannot be possessors, so all those entities can be ruled out from the set of possible hyperonyms for every unknown concept seen in genitive case somewhere in the corpus.

Like Camille, this system does not return a single hyperonym but the set of plausible hypotheses.

Discussion

Existing work on Ontology Refinement is scant and disperse. The approaches described in this section not only use different corpora and different ontologies for training and evaluating, as shown in Table 3.5, but also the evaluation metrics they use are different. This shows the need of a stable framework on which to train and test future algorithms that attempt this task.

Concerning the possible improvements of existing approaches, neither Hearst [1998] nor Kietz et al. [2000] solved the problem that word patterns are not very reliable, and have to be manually revised. On the other hand, the approaches by Hastings [1994] and Hahn and Schnattinger [1998] are somewhat more reliable, but their approaches need many rules and scripts that they use for the classification, which contain selectional restrictions and some common sense knowledge, and they do not describe any way to learn them. Their main drawback is the difficulty of creating this set of rules.

3.5 Summary

A Lexical Knowledge Base (LKB) is a repository that contains words and information about them. This information can be of several kinds, such as morphological (how a word is inflected), syntactic (which is the category of the word and how it combines with others to make complex constituents), semantic (what is the meaning of the word), multilingual (what is the form of that word in other languages), etc.

A particular case of LKB are conceptual lexical ontologies, which are semantic networks that relate concept with semantic relationships. In a conceptual ontology, the nodes of the network are concepts (which can be represented by the sets of synonym words that express those concepts), and the arcs are the relationships between them. Some of the possible relationships that can be encoded are *hyperonymy*, that relates a concept with the ones that are more general and its opposite relation *hyponymy*, which relates a concept with the ones whose meaning is more specific. With these relationships, it is possible to establish a taxonomy of concepts where the more general concepts are located at the top, and the more specific concepts are located at the leaves. A popular conceptual ontology widely used in the field of NLP is WordNet.

The information encoded in conceptual ontologies can be acquired with automatic methods, using different procedures. The existing work is very varied. So, ontology learning systems can require validation by a judge or be fully automatic; they can learn from different kinds of input data, such as Machine-Readable Dictionaries (MRDs) or unrestricted text; they can either start working with words and group them in concepts (*descriptive approach*) or start with the concepts and attach the words to the concepts (*prescriptive approach*); they can create an ontology from scratch or modify slightly an already existing ontology; and they may require different degrees of supervision and use different techniques and learning methods.

A particular case of Ontology Learning happens when an existing ontology is adapted to the necessities of some application or user. This may involve deleting the unwanted concepts and adding new relevant concepts, which might not be present in the initial ontology, in their appropriate places. Current approaches to add new concepts to an existing ontology include using contextual regular expressions that express hyperonymy in natural language, or classifying words according to the context in which they occur.

Chapter 4

Distributional Semantics

This chapter describes the hypotheses posed by the Distributional Semantics model and their implications. Section 4.2 includes a brief account of the computational tools motivated by these hypotheses and their applications for Natural Language Processing. Section 4.3 describes some experiments to show how these tools hold when applied to language processing. Finally, Section 4.4 ends the chapter with a general discussion.

4.1 Introduction

When we hear a new word that we did not know before, we can guess some of its meaning from the way it is used. For instance, in sentence (6a) we can guess that Fury is an animal, most likely a horse, because the verb *to gallop* usually selects a horse as its subject (except in metaphorical contexts); however, in sentence (6b) we can imagine that *Fury* is something that can be displayed on television, e.g. a film or a TV show.

(6) a. Fury galloped toward the forest

b. They show Fury today at channel three.

An English speaker can guess, even unconsciously, the meaning of the unknown word *Fury* in these two sentences, because of the knowledge that only certain kinds of *entities* are able to gallop (and hence can appear, in language, as the subject of the verb *to gallop*), and that only other *entities* can be shown on TV (and therefore can appear as the direct object of the verb *to show*). Furthermore, all the kinds of entities that are able to gallop are semantically related in some way; and all the kinds of entities that can be shown on TV have also some semantic similarities. The Distributional Semantics model explains this phenomenon by assuming that the meaning of a word is strongly related to the contexts in which it is used. Citing Church et al. [1991] (from Resnik [1993]):

> Our approach has much in common with a position that was popular in the 1950s. It was common practice to classify words not only on the basis of their meanings but also on the basis of their co-occurrence with other words. Running through the whole Firthian tradition, for example, is the theme that "You shall know a word by the company it keeps" [Firth, 1957b]. Harris' "distributional hypothesis" dates from the same period. He hypothesized that "the meaning of entities, and the meaning of grammatical relations among them, is related to the restriction of combinations of these entities relative to other entities" [Harris, 1968, pg. 12].

In summary, the Distributional Semantics model starts with the following hypothesis.

> **Hypothesis 1.** The meaning of a word w is highly correlated to the contexts where w appears [Rajman and Bonnet, 1992].

From this assumption, it is possible to develop statistical computational tools for calculating similarities in word meanings, which have been applied to Information Retrieval [Rajman and Bonnet, 1992, Salton, 1989], Text Summarisation [Lin, 1997], word-sense disambiguation [Yarowsky, 1992, Agirre et al., 2000a], and word clustering [Lee, 1997, Faure and Nédellec, 1998].

This chapter starts with some commonly agreed definitions of the semantic relations that are more relevant for characterising word meaning: hyponymy and synonymy. Next, the procedures followed in the Distributional Semantics model will be described, and finally tested with a brief empirical evaluation.

4.1.1 A definition of hyponymy and synonymy

Word meanings can be related through many possible semantic relationships, such as PART-OF, which relates a concept with its constitutive parts; or TELIC, that relates a concept with its purpose or finality (e.g. the purpose of a cigarette is to smoke). This subsection discusses two of the most studied semantic relationships: hyponymy and synonymy.

Hyponymy is a semantic relationship which relates a concept with more general concepts, such as *horse* with *animal*. It can be defined in the following way:

> **Definition 1a. Hyponymy** is a relation of meaning inclusion between linguistic expressions. A is a hyponym of B if B is true for any concept or instance x whenever A is true for x.
>
> **Hyperonymy** is the inverse relation to hyponymy.

According to this definition, if a property P is true for every instance x in a class A, then P is a hyperonym of A. For example [Resnik, 1993], every single QUEEN is a WOMAN, and therefore QUEEN is a hyponym of WOMAN. This implies that any utterance about a queen x entails the same utterance where x is referred as being a woman, e.g. (7a) entails (7b).

(7) a. The Prime Minister honoured the queen with his presence.

b. The Prime Minister honoured the woman with his presence.

This lead us to the definition of hyponymy in terms of interchangeability of linguistic expressions:

> **Definition 1b.** A is a **hyponym** of B if and only if for every sentence S containing A, then S entails the same sentence with A substituted for B, $S[A/B]$ [Lyons, 1961].

As Resnik [1993] notes, a possible difficulty of using this definition in a theory of lexical semantics is that this "kind-of" relationship is not clearly linguistic; on the contrary, it may be factual.

> So, for example, even if it were true that anything which is a dog is hairy, and that the ordinary language user would be prepared to say "x is hairy" given that "x is a dog", one might not want to say that there is a hyponymic relationship between the words *canine* and *hairy*. On the one hand, one could argue that being hairy is a definitional aspect of being a dog, in which case the

> relationship does belong within the domain of lexical semantics; but on the other hand, one could argue against this position, saying that the hairiness of dogs is accidental and therefore a matter of factual knowledge and not definition.

However, Resnik concedes that it is also not entire clear that the "kind of" relationship is non-linguistic; in the example of the queen and the woman, the inheritance of the gender is "inextricably wrapped up in the meanings of the lexical items and not just accidental facts about the world". In the same way, in the example above, if it were true that every single dog is hairy, and every speaker agreed that it can be inferred "x is hairy" from the statement "x is a dog", then it can be argued that the concept *hairy* or *hairy_animal* has to be a hyperonym of *dog*.

Resnik [1993], describing the lexical ontology WordNet [Miller, 1995], provides another possible definition of hyponymy.

> **Definition 1c.** If a and b are distinct hyponyms of c, and α, β and γ are the sets of plausible entailments generated by virtue of membership in each respective class, then $\gamma \subset \alpha$, $\gamma \subset \beta$ and $\alpha \neq \beta$ [Resnik, 1993, pg. 24].

Definition 1c, albeit true when referring to WordNet, does not account for the fact that a taxonomy of concepts can be non-monotonic, i.e. a particular hyponym may not have some property of its hyperonym[1]. For example, most birds fly, and therefore any English speaker will include "x flies" as a plausible entailment of "x is a bird". On the other hand, penguins are some kind of birds that do not fly, and therefore "x flies" is not an entailment of "x is a penguin".

In definition (1c), Resnik uses the implication in the sense used by Lyons –meaning a *common-sense plausible* implication:

> Having said "x is a piece of fruit", the ordinary speaker of English can reasonably be expected to agree that "x is an item of groceries or foodstuff" is also true, as well as any additional facts plausibly entailed by that statement on either linguistic or non-linguistic grounds.

As a final remark, according to Wierzbicka [1984], the hyponymy relationship may represent at least five different relationships, from which two salient ones are IS_A_KIND_OF and IS_USED_AS_A_KIND_OF; Wierzbicka calls them *taxonomic* and *functional*, respectively. For example, a WRITTEN_AGREEMENT is a kind of LEGAL_DOCUMENT, and it is used as a kind of AGREEMENT. According to the definitions, it is a hyponym of both of them, but the semantic relationships that relate it to the two classes are different.

For our study, any of the definitions stated here can be considered as valid. Because we are using WordNet, and we are not going to use any non-monotonic property over the concepts, definition (1c) can also be considered valid.

The second lexical relationship described in this section is **synonymy**, which relates concepts that convey the same meaning. In Miller [1995], synonymy is characterised as a matrix that relates word meanings to word forms (see Table 4.1). Word forms are typically sequences of characters delimited by spaces. However, in some contexts special symbols may be considered words and, as Resnik [1993] points out, provision must be made for multi-word expressions. Word meanings refer to "the lexicalised concept that a form can be used to express" [Miller, 1995]. A particular example with some concepts and word forms is shown in Table 4.2.

[1]For the discussion on the monotonicity of WordNet, please refer to Miller [1998]

Word Meanings	Word forms f_1	f_2	f_3	...	f_n
m_1		×			
m_1		×			×
m_1					
...					
m_n			×		

Table 4.1: Schematic representation of a lexical matrix, from Resnik [1993].

Word Meanings	Word forms horse	heroin	junk	debris	...
horse, Equus sp.	×				
horse, heroin (drug)	×	×	×		
junk (Chinese boat)			×		
debris, detritus			×	×	
...					

Table 4.2: Example of lexical matrix, showing some words and the concepts they lexicalise.

Here, {horse, heroin, junk} is the set of synonyms (also called synonym set or, abbreviated, *synset*) that represents the concept *heroin*.

Spark-Jones [1964] characterises synonymy as follows. First, as a background assumption, sentences are taken to have a property she calls a *ploy*, which is the way in which it is employed –for example, *Shut up* and *Keep quiet* have the same ploy. Given a sentence S and one of the word positions in S, a row is defined as a set of words that can appear in that position without changing the ploy of S.

We can define synonym words as words that convey the same meaning. Therefore, we can also write the definitions parallel to (1a) and (1b), using the fact that synonym words must be interchangeable in every context.

> **Definition 2a. Synonymy** is a relation of meaning identity between linguistic expressions. A and B are synonyms if and only if B is true for a concept or instance x whenever A is true for x and vice versa.
>
> **Definition 2b.** Two word forms w_1 and w_2 are **synonyms** if and only if for every sentence S containing A, then S entails $S[A/B]$, and for every sentence T containing B, then T entails $T[B/A]$.

Corollary 2c comes straightforwardly from definition 2b. If two word forms w_1 and w_2 are exchangeable in every sentence where any one of them appears, then they can be used in exactly the same contexts in language:

> **Corollary 2c.** If two word forms w_1 and w_2 are synonym, then they can appear in exactly the same contexts, preserving the truth value.

Finally, we can define synonymy in terms of hyperonymy as in the following definition. It can be seen that, if we use definition (2d), then (2a) and (2b) can be derived from (1a) and (1b).

Definition 2d. Two word forms w_1 and w_2 are synonyms if and only if both w_1 is a hyperonym of w_2 and w_2 is a hyperonym of w_1.

However, it is very rare to find words which are absolute synonyms. Synonyms usually have small denotations that differentiate every one of them from the others. In some cases there are small differences in meaning; in others, that difference may be only stylistic, such as dialectal variations: geographical variations (such as elevator/lift or the Spanish variation melocotón/durazno); historical variations (e.g welkin/sky); or depending on the cultural level of the speaker or the informality of the language. Even in this last case it may be the case that two synonyms are not fully interchangeable; for example, sentences (8a), (8b) and (8c) convey exactly the same message, but the last one sounds weird, because of the mixture of formal, slang and scientific words.

(8) a. The policeman enjoined the drug dealer to turn over the heroin [formal].

b. The cop ordered the pusher to pass on the horse [slang].

c. The cop enjoined the pusher to turn over the diacetylmorphine [weird].

Some semanticists argue that the denotational meaning of a word is fully realised in contexts. As Firth [1957a, pg. 7] says, "The complete meaning of a word is always contextual, and no study of meaning apart from a complete context can be taken seriously", a theory agreed also by Edmonds and Hirst [2002]. Under this premise, it is rarely that two words have exactly the same meaning and are exchangeable in every possible context. Edmonds and Hirst argue that many words are not absolute synonyms, but near-synonyms (also called *plesionyms*). They discuss the topic in the following passage:

> Absolute synonymy, if it exists at all, is quite rare. Absolute synonyms would be able to be substituted one for the other in any context in which their common sense is denoted with no change to truth value, communicative effect, or 'meaning' (however 'meaning' is defined). Philosophers such as Quine (1951) and Goodman (1952) argue that true synonymy is impossible because it is impossible to define, and so, perhaps unintentionally, dismiss all other forms of synonymy. Even if absolute synonymy were possible, pragmatic and empirical arguments show that it would be very rare. Cruse (1986, p. 270) says that "natural languages abhor absolute synonyms just as nature abhors a vacuum", because the meanings of words are constantly changing. More formally, Clark (1992) employs her Principle of Contrast, that "every two forms contrast in meaning", to show that language works to eliminate absolute synonyms. Either an absolute synonym would fall into disuse or it would take on a new nuance of meaning. At best, absolute synonymy is limited mostly to dialectal variation and technical terms, but even these words would change the style of an utterance when intersubstituted.

Table 4.3 contains several of the ways that near-synonym words may vary in their meaning. In most cases, this difference is subtle.

For the purposes of this work it can be considered that near-synonyms behave as true synonyms, an approach that was taken when designing WordNet. In the rest of this work, the word *synonym* will be used to refer to near-synonyms or plesionyms. If, in any occasion, it is necessary to make the distinction, synonyms that have exactly the same meaning will be called *absolute synonyms.*

Other definition of synonymy that we can consider, for practical reasons, is the following definition (2e), provided by Resnik [1993]. As can be seen, it is not precise (it is not clear what *representative* means), but it is useful enough for working with WordNet.

Variation	Example
Abstract dimension	seep : drip
Emphasis	enemy : foe
Denotational, indirect	error : mistake
Denotational, fuzzy	wood : forest
Stylistic, formality	heroin : horse
Stylistic, force	ruin : annihilate
Geographical origin	elevator : lift
Time of utterance	welkin : sky; betwixt : between
Express attitude	skinny : slim : thin : slender
Emotive	daddy : dad : father
Collocational	task : job
Selectional	pass away : die
Subcategorisation	give : donate

Table 4.3: Examples of near-synonym variation. Most are taken from Edmonds and Hirst [2002].

Definition 2e. Two word forms w_1 and w_2 are **synonyms** if and only if there is a "representative" set of sentences $\{S_j\}$, containing either w_1 or w_2, such that if S_j entails σ, then $S_j[w_1/w_2]$ and $S_j[w_2/w_1]$ also entails σ [Resnik, 1993].

For example, *board* and *plank* are considered *synonyms* in the semantic domain of carpentry, although there may be subtle differences between them.

4.2 The Distributional Semantics model

We can formulate the Distributional Semantics assumption [Rajman and Bonnet, 1992, Church et al., 1991] with the hypothesis 1, repeated here for convenience of the reader, and with some corollaries that we can derive from it:

Hypothesis 1. The meaning of a word form w is highly correlated to the contexts where w appears.

Corollary 3a. If two word forms w_1 and w_2 can appear in exactly the same contexts, their meaning must be nearly the same.

Corollary 3b. If two word forms w_1 and w_2 can appear in "similar" contexts, their meaning must also be "similar".

Corollary 3c. It is possible to guess the meaning of w by a corpus study of the contexts in which it is used.

Corollary 3d. It is possible to guess in which contexts w can be used if we know its meaning.

Note the similarities between the corollaries and definition (2b) of synonymy (and its corollary (2c)) given above. (2c) stated that two words share the same meaning then they can be interchanged in every sentence where one of them appears and the truth value of the sentence is the same. Corollary (3a), in contrast, does not require the additional constraint that the truth value of the sentence must be the same.

On the other hand, these hypothesis and corollaries do include some terms whose meaning would be desirable to clarify, such as *nearly*, *highly*, *guess* or *similar*. This section describes how these corollaries can

be applied to NLP applications by using statistical models, and how to use these models to show how to quantify the similarity of meanings.

In order to test the hypothesis and its corollaries, we need a distance metric between word meanings, a distance metric between contexts, and the tools required to measure them, so we can then prove that the values of both metrics are correlated.

4.2.1 Distance metric between contexts

By assuming hypothesis 1 above, the tools used in the Distributional Semantics model try to parametrise different characteristics of the contexts of words, in order to calculate semantic distances between them. These can be classified according to the following criteria:

1. Metrics.

2. Contextual relationships between the words.

Metrics

The simplest context co-occurrence metric that can be calculated is a multi-dimensional vector of frequencies of co-occurrence between words. For example, if we want to capture the context of a word w, we can describe it with the vector of words that appear around w, together with their frequencies of appearance. The context size can be defined as a window of n words at each side of w, or as the whole sentences where w appears. Table 4.4 shows the context words (including punctuation symbols) and their frequencies that were collected from every sentence that contained the word *man*, from a corpus of documents collected from Internet, all of which contained the word *man*.

Using the vectors of frequencies, from the perspective of Information Theory, we can calculate the information that a context word provides about a concept. Let us suppose that we have a corpus with N words, and that words w_i and w_j appear $freq_i$ and $freq_j$ times in the corpus, respectively. Then, we can estimate their probabilities of appearance of w_i and w_j as

$$p(w_i) = \frac{freq_i}{N} \tag{4.1}$$

$$p(w_j) = \frac{freq_j}{N} \tag{4.2}$$

In a similar way, we can define $p(w_i, w_j)$ as the probability of seeing the two words together, and $p(w_j|w_i)$ as the probability of seeing w_j given that we have seen w_i.

Entropy is an Information Theory measure defined as the expected value of $-log\ p(x)$ for a variable X. It can be considered as a measure of how uncertain we are about the value that X will have, or the average of amount of information received when the value of X is observed.

$$\mathrm{H}(X) = E[-log\ p(x)] = -\sum_x p(x) log\ p(x) \tag{4.3}$$

Other interesting metric is the *conditional entropy* $\mathrm{H}(X|Y)$, which measures the uncertainty of X, given

Word/Token	Frequency	Word	Frequency
,	23696	has	449
-	3964	1	441
:	3774	der	436
»	3349	–	428
's	2868	2	396
was	2099	when	389
is	2066	und	376
;	1990	Re	376
!	1609	will	373
'	1467	0	373
I	1383	which	345
?	1208	John	331
de	1028	would	331
not	970	il	327
be	739	been	323
who	687	were	318
do	631	que	315
had	615	can	296
have	563	years	291
old	524	time	290
o	516	2000	288
]	500	where	268
[	483	first	263
e	477	year	263
are	471	die	260
man	458	How	260

Table 4.4: Tokens that co-occur with the word *man* in the same sentence, and their frequencies of appearance.

that the value of Y is known:

$$\mathrm{H}(X|Y) = E[-log\ p(x|y)] = -\sum_{x,y} p(x,y) log\ p(x|y) \tag{4.4}$$

The *Mutual Information* of two random variables X and Y represents the information that is common between them. It can be interpreted as the information that Y provides about X: the decrease in the uncertainty of X that happens when we learn Y's value.

$$\mathrm{I}(X,Y) = \mathrm{H}(X) - \mathrm{H}(X|Y) \tag{4.5}$$

Note that mutual information is symmetrical, i.e., $\mathrm{I}(X,Y) = \mathrm{I}(Y,X)$.

$$\begin{aligned} \mathrm{I}(x,y) &= E[-log\ p(x)] - E[-log\ p(x|y)] \\ &= E\left[-log\frac{p(x)}{p(x|y)}\right] \\ &= E\left[log\frac{p(x,y)}{p(x)p(y)}\right] \\ &= E\left[log\frac{p(y,x)}{p(y)p(x)}\right] \end{aligned} \tag{4.6}$$

Intuitively, $p(x,y)$ is the probability of seeing x and y together. However, if the probabilities of appearance of the words alone $p(x)$ and $p(y)$ are high, then the probability that x and y are seen together by chance $p(x)p(y)$ can also be high. In order to correct the metric, $p(x,y)$ is divided by $p(x)p(y)$. If both probabilities are the same, by taking the logarithm, the mutual information will be zero, i.e., there is no information in common between x and y.

Mutual Information was used by Hindle [1990] and Resnik [1993] to capture selectional constrains and other syntactic relationships among words. As an example, Table 4.5 shows the words that the verb *to drink* selects as direct objects, and their mutual information. The score is higher if the pair has been seen with higher frequency in a six-million-word corpus. In this way, it is possible to obtain, for each verb, a multi-dimensional vector with selectional preferences. These vectors can be used to classify an unknown verb, by comparing the vector corresponding to this unknown verb with all the other verbs' vectors. They can be used as well, for example, for clustering these verbs according to their selectional preferences.

If we collect these contextual vectors for several words, we shall find that some of the context words are equally frequent in all the vectors, such as the punctuation symbols and the closed class words (prepositions, conjunctions, pronouns, etc.). Therefore, it would be desirable to use statistical methods to discover which of the words are equally frequent in every context, and which ones are salient only in the contexts of specific words. To do this, there are several statistical tests available. Some of the most used are the t-score, tf·idf and χ^2.

Let us suppose that we have two concepts w_i and w_j, and for each one of them we have collected a list of context words and frequencies

$$\{< c_1, freq_{i1} >, < c_2, freq_{i2} >, ..., < c_t, freq_{it} >\}$$

score	verb	object
12.34	drink	bunch (of) beer
11.75	drink	tea
11.75	drink	Pepsi
11.75	drink	champagne
10.53	drink	liquid
10.20	drink	beer
9.34	drink	wine
7.65	drink	water
5.15	drink	anything
2.54	drink	much
1.25	drink	it
1.22	drink	SOME AMOUNT

Table 4.5: Mutual information between verbs and objects; for the verb *to drink* [Hindle, 1990, from Resnik [1993]]

t	strong w	powerful w	w
12.42	161	0	showing
11.94	175	2	support
10.08	550	68	,
9.97	106	0	defense
9.76	102	0	economy
...	...	...	...
-4.91	0	24	post
-5.23	1	28	of
-5.37	3	31	minority
-5.60	1	32	figure
-7.44	1	56	than

Table 4.6: Values of the t-score when comparing words preceded by *strong* and *powerful*, from Church et al. [1991]. The second column shows the number of times that the word appeared after *strong*, and the third column shows the number of times that it appeared after *powerful*.

By calculating the frequencies of w_i and w_j in the whole corpus, we can estimate the probabilities $P(w_i)$, $P(w_j)$, $P(c_k)$, $P(w_1, c_k)$, $P(w_2, c_k)$, $P(c_k|w_1)$, $P(c_k|w_2)$, etc., and the mutual informations $\mathrm{I}(w_1, c_k)$ and $\mathrm{I}(w_2, c_k)$.

If we want to discover which context words appear more frequently in the context of w_i, and which ones appear more frequently in the context of w_j, a possible test is the t-score [Church et al., 1991]:

$$t = \frac{P(c|w_i) - P(c|w_j)}{\sqrt{\sigma^2 \cdot P(c|w_i) + \sigma^2 \cdot P(c|w_j)}} \tag{4.7}$$

Table 4.6 shows the values of the t-score when comparing words preceded by the adjectives *strong* and *powerful*. Some of them, such as *support* or *economy*, show a strong preference to be modified by *strong*, while others such as *minority* or *post* show a preference for *powerful*. The higher the value of the t-score, the more confidence we have that this difference is real and not due to mere chance. As an example, a t value of 1.65 give us 95% confidence that there is a preference for one of the two adjectives.

The **tf·idf** is an alternative measure that is used mostly for Information Retrieval applications [Salton,

1989, chapter 9]. The following is a small variation of the original algorithm in order to be applied to words and contexts. Let us suppose that we have some words whose contexts we want to study, such as {*strong, powerful*}. According to the model tf·idf, the weight of a context word c_k in the context of word w_i is calculated as

$$weight_{ik} = tf_{ik} \cdot log_2 \frac{N}{n} \tag{4.8}$$

where tf_{ik} is the frequency of c_k in the context of w_i; N is the number of words; and n is the number of words such that c_k appears in their context at least once.

Note that if a context word appears in the context of every word w_i, then the value of the logarithm is zero, and therefore its weight will be zero. In this way, closed class words and other generic words will not receive a weight.

Finally, other test that can be applied to calculate words weights is the χ^2. Under this model, let us suppose again that we have a set of words $\{w_1, ..., w_N\}$ whose contexts we want to study; and a set of context words $\{c_1, ..., c_M\}$. Let's call $freq_{ik}$ the frequency of word c_k in the context of w_i. Then, the expected mean m_{ik} is defined as

$$m_{ik} = \frac{\sum_i freq_{ik} \cdot \sum_k freq_{ik}}{\sum_{i,k} freq_{ik}} \tag{4.9}$$

The weight for context word c_k in the context of w_i is then

$$weight_{ik} = \begin{cases} \frac{(freq_{ik} - m_{ik})}{m_{ik}}, & \text{if } freq_{ik} > m_{ik} \\ 0 & \text{otherwise} \end{cases} \tag{4.10}$$

The aim of all these models is to assign, to each word in the context of a concept c, a weight that takes into consideration the frequencies of that word in every other concept's context. So, if two context words have appeared in the context of a word with different frequencies, but neither of them appears in the context of any other synset, they both will score the maximum value of the weight, because they both are maximally supporting that word.

Once every context word has received a weight in the context of each of the concepts, we can compare the contexts in several ways. Yarowsky [1992] and Agirre et al. [2000a] use the **dot product** of the context vectors. If $weight_{ik}$ is the weight of context word k in the context of the concept c_k, then the similarity of concepts w_i and w_j is

$$Similarity_{ij} = \sum_{k=0}^{n} weight_{ik} \cdot weight_{jk} \tag{4.11}$$

Other similarity metric that has been also widely used for Information Retrieval is the **cosine of the angle** of the two vectors of words:

$$\begin{aligned} sim(w_i, w_j) &= \frac{w_i \cdot w_j}{|w_i| \times |w_j|} \\ &= \frac{\sum_{k=1}^{t} w_{ik} \times w_{jk}}{\sqrt{\sum_{k=1}^{t} w_{ik}^2} \times \sqrt{\sum_{k=1}^{t} w_{jk}^2}} \end{aligned} \tag{4.12}$$

Other similarity metric, again taken from Information Theory, is the **relative entropy**, also called

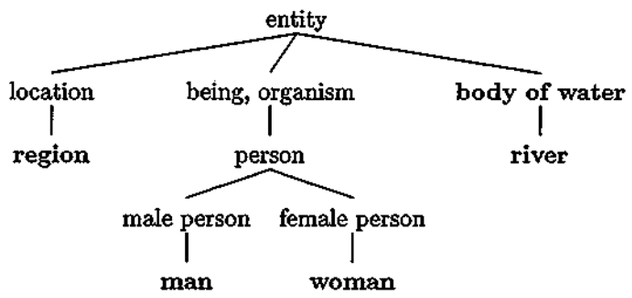


Figure 4.1: Example of taxonomy (extracted from WordNet).

Kullback-Leibler distance. Given two probability distributions $p(x)$ and $q(x)$, the relative entropy between p and q is defined as

$$D(p||q) = \sum_{x \epsilon \mathcal{X}} p(x) log \frac{p(x)}{q(x)} \tag{4.13}$$

It is not really a distance metric, as it is not symmetric, because $q(x)$ only appears in the denominator; but it provides an estimate of the inequality of both distributions. Hence, if we divide the weight of every context word $weight_{ik}$ by the sum of all weights $\sum_k weight_{ik}$, to make them all add up 1, two context vectors can be compared using this metric.

Contextual relationships

The contextual vectors shown in Tables 4.4 and 4.6 show co-occurrence of words in the same context; however, in other occasions, it may be useful to collect only words which bear some kind of relationship. So, for example, Table 4.5 showed the direct objects that the verb *to drink* can select [Hindle, 1990]. The following is an account of the kinds of contextual information that can be considered:

- Co-occurrence: the words that appear in the same context; if the context window is of width 1, then the two words have to be consecutive, and in this special case only bigrams can be studied.
- Syntactic relationships: for example, the verbs that select some nominal concept as a subject; or the nouns that a verb can select as subject. In principle, vectors can be calculated for every possible syntactic relationship that any category accepts.
- Semantic relationships: for example, we may only include in the vector of a word w the context words that are related through semantic relationships (e.g. meronymy or holonymy, antonymy, etc.) with w in a semantic network.

4.2.2 Distance metrics between meanings

We already have several measures of similarity between word contexts; we also need, in order to test empirically the Distributional Semantics hypothesis, a measure of semantic similarity between concepts.

To calculate the semantic similarity between two concepts, we will make use of the WordNet ontology [Miller, 1995]. This ontology has been used in many ways to calculate different metrics of similarity.

Some of the metrics take into consideration the number of hops between two nodes in the WordNet semantic network. This is the case of the distance metric described by Hahn and Schnattinger [1998]. However, these metrics have the problem that, in WordNet, different hyperonymy links really represent different semantic distances. As can be seen in the WordNet extract in Figure 4.1, the number of hops between *man* and *woman* and between *person* and *region* are the same: four hops. This is due to the fact that the links that are nearer the top of the ontology separate concepts that have a very different meaning, such as locations, life beings and artifacts; while, as we proceed down the ontology, these links join very similar concepts (such as *widow* and *war widow*; or *maid* and *girl*).

Therefore, hyperonymy links should be weighted in order to capture this fact. Resnik [1999] described how to calculate the Information Content for a WordNet synset s as the negative log likelihood $-log\ p(s)$, where p(s) is the probability of finding a word that belongs to that synset or any of its hyponyms. If the ontology is rooted on just one synset, i.e. if every synset in the whole ontology is rooted in some synset r, then r will have a probability of 1, and an Information Content of 0. The less frequent a concept is, the more informative it is.

The similarity of the two concepts s_1 and s_2 can be defined as the amount of Information Content they share, i.e. the Information Content of the most informative common hyperonym in the ontology c. We can calculate the following metrics:

- $I(c)$ represents the amount of Information Content of the common ancestor c.
- $I(s_1) - I(c)$ is the amount of Information Content that s_1 adds to the information of c.
- $I(s_2) - I(c)$ is the amount of Information Content that s_2 adds to c.

There are many possible ways of combining these values to calculate a similarity metric between two concepts; one of them can be calculated as follows: the amount of information they share divided by the maximum of the information contents that the particular concepts have:

$$\frac{I(c)}{max(I(s_1), I(s_2))} \tag{4.14}$$

Therefore, if two concepts are synonyms (i.e. if they belong to the same synset), their similarity will be 1; on the other hand, if the most specific common ancestor is the root of the ontology, their similarity will be 0.

To calculate the Information Content of the WordNet synsets, the SEMCOR corpus was used, because every noun in that corpus is annotated with the WordNet synset corresponding to the sense with which that word is used. The steps are the following:

1. For each synset, the frequencies of appearance of the synset words and its hyponyms' are calculated.
2. By dividing these frequencies between the total number of nouns in the corpus, it is possible to estimate their probabilities, and calculate the Information Content metric.

After performing these steps, for example, the similarity between *man* and *woman*, in the ontology from

Figure 4.1, is the following value:

$$\begin{aligned} & \frac{I(person)}{max(I(man), I(woman))} \\ = & \frac{1.8984}{max(5.4098, 5.5671)} \\ = & 0.3410 \end{aligned} \tag{4.15}$$

4.3 Empirical analysis

For the empirical evaluation, five WordNet synsets were chosen: *man*, *woman*, *river*, *body of water* and *location*. As shall be seen, the preliminary results with these five synsets are extremely good, although for a more thorough evaluation these experiments should be done with more sets of synsets chosen randomly. Table 4.7 shows the similarity values between the five chosen WordNet synsets. The synsets that are most related are *river* and *body of water* (0.840); next, *man* and *woman* (0.341); all the remaining pairs have a similarity below 0.14.

The aim of the experiments was to test the correlation between the meaning of the concepts and the contexts in which they appear. For the first experiment, the following steps were performed:

1. For each of the five synsets, a collection of documents were automatically collected from Internet, containing the words from that synset.

2. For each of the collections, all the words that appeared in the context of any of the synset words were collected in a vector of words and frequencies.

3. For each of the vectors of words and frequencies, the other four vectors were taken as contrast set in order to change the frequencies into weights, using the χ^2 metric.

4. The context word distributions were compared using the Kullback-Leibler distance or relative entropy.

The results are shown on Table 4.8. If we calculate the correlation between the similarities of the synsets and the similarities of the contexts, we obtain the value 0.9739, which means that this correlation is nearly perfect.

Next, instead of collecting words that co-occurred in the same contexts, only the verbs for which these synsets were in subject position were collected for the vectors of words. The remaining steps were the same: the χ^2 metric was used to calculate the weights, and the Kullback-Leibler distance to measure a distance between the vectors. The results are shown in Tables 4.9. The correlation with the semantic distance in this case was 0.9697.

Still two more experiments were done: by only collecting the verbs for which these nouns were in object position; and the words (adjectives and determiners) that modified these synsets. The results are shown in 4.10 and 4.11. The correlations with the semantic distances are 0.9739 and 0.9752.

It must be noted that this experiment requires that the vectors of context words are large enough; if the set of collected documents is very small, then the results will not be so good, as it suffers from the sparse data problem. This experiment was performed with several hundred documents for each of the synsets.

It can be argued that the correlation depends of the similarity measure chosen both between the WordNet synsets and between the contexts. No experiment has been performed using different distance metric (e.g.

	location	man	woman	body of water	river
location	×				
man	0.138	×			
woman	0.134	0.341	×		
body of water	0.135	0.113	0.134	×	
river	0.113	0.135	0.113	0.840	×

Table 4.7: Semantic similarity between five concepts taken from WordNet. The higher the number, the higher their similarity.

	location	man	woman	body of water	river
location	×				
man	$8.8 \cdot 10^{-7}$	×			
woman	$8.8 \cdot 10^{-7}$	$25.8 \cdot 10^{-7}$	×		
body of water	$8.6 \cdot 10^{-7}$	$7.2 \cdot 10^{-7}$	$7.5 \cdot 10^{-7}$	×	
river	$10.1 \cdot 10^{-7}$	$8.0 \cdot 10^{-7}$	$5.8 \cdot 10^{-7}$	$242.7 \cdot 10^{-7}$	×

Table 4.8: Similarity of the context of synset pairs, using the χ^2 estimate and the Kullback-Leibler distance.

	location	man	woman	body of water	river
location	×				
man	$21.7 \cdot 10^{-5}$	×			
woman	$9.1 \cdot 10^{-5}$	$38.2 \cdot 10^{-5}$	×		
body of water	$23.2 \cdot 10^{-5}$	$14.7 \cdot 10^{-5}$	$7.5 \cdot 10^{-5}$	×	
river	$15.8 \cdot 10^{-5}$	$17.7 \cdot 10^{-5}$	$7.2 \cdot 10^{-5}$	$416.0 \cdot 10^{-5}$	×

Table 4.9: Similarity of the context of synset pairs, using the χ^2 estimate and the Kullback-Leibler distance.

	location	man	woman	body of water	river
location	×				
man	$14.1 \cdot 10^{-5}$	×			
woman	$8.8 \cdot 10^{-5}$	$29.9 \cdot 10^{-5}$	×		
body of water	$19.0 \cdot 10^{-5}$	$7.6 \cdot 10^{-5}$	$5.1 \cdot 10^{-5}$	×	
river	$16.4 \cdot 10^{-5}$	$12.9 \cdot 10^{-5}$	$6.7 \cdot 10^{-5}$	$229.6 \cdot 10^{-5}$	×

Table 4.10: Similarity of the context of synset pairs, using the χ^2 estimate and the Kullback-Leibler distance.

	location	man	woman	body of water	river
location	×				
man	$5.8 \cdot 10^{-6}$	×			
woman	$4.8 \cdot 10^{-6}$	$22.4 \cdot 10^{-6}$	×		
body of water	$5.8 \cdot 10^{-6}$	$5.1 \cdot 10^{-6}$	$4.2 \cdot 10^{-6}$	×	
river	$5.4 \cdot 10^{-6}$	$7.2 \cdot 10^{-6}$	$3.8 \cdot 10^{-6}$	$217.0 \cdot 10^{-6}$	×

Table 4.11: Similarity of the context of synset pairs, using the χ^2 estimate and the Kullback-Leibler distance.

using the tf-idf metric for calculating the weights, or using the dot product of the cosine similarity for comparing the vectors). But it is a very important fact is that there exist at least one metric which is theoretically-grounded for which the correlation is so extremely high.

4.4 Summary

This chapter contains an overview of the different models for Distributional Semantics, a framework that is based on the assumption that the meaning of a word or concept is highly correlated with the contexts in which it appears. The chapter started with an introduction to the two semantic relationships that are more relevant to the work described in this thesis, hyponymy and synonymy, that will appear all through this work.

Some of the most widely used metrics to weight the context words, in order to find which are more relevant in the context of a given concept, are the t-score, tf-idf, and χ^2. These can be used to generate, for a given concept, a vector of context words, each with an associated weight.

Next, these vectors of context words and weights can compared, so as to calculate a measure of similarity between two concepts. Some of the possible metrics for comparison are the scalar product of the context vectors; the cosine of the angle that the vectors form; or the Kullback-Leibler distance. These context vectors have been successfully applied to Information Retrieval, Text Summarisation and word-sense disambiguation, among other applications.

Finally, an empirical experiment is described, with a few concepts taken from WordNet. The correlation between the similarity of the concepts in the network and the similarities of their contexts is very high, always around 0.97.

The Distributional Semantics model has proven useful for many applications, including Information Retrieval, Text Summarisation, document classification, word-sense disambiguation or concept clustering, amongst others. The following drawbacks are usually present in these models:

Sparse data. This is a problem that usually arises in corpus-based algorithms. Because of the versatility of human language, it is unlike that any corpus will contain every possible linguistic phenomenon.

Smoothing methods try to ameliorate the problem. A commonly-used smoothing method consists in incrementing the frequency of every word in the vector representations.

Computational expense. Distributional Semantics typically require large quantities of storage space and execution time, as they require the processing of large corpora.

Homonyms. There are some words that, although they have different meaning, share the same lexical form, such as *junk* meaning "a Chinese boat" or "rubbish". With a distributional method, the contexts of *junk* should be separated according to its meaning, in order to collect the word frequencies separately. Some words are specially polysemous: the verb *to keep* has 22 distinct senses in WordNet; and the verb *to give* has 44 senses. A word-sense disambiguation procedure may be necessary in order to collect the word vectors for each of the WordNet synsets separately.

Chapter 5

Distributional Semantics Applied to LKA

This chapter introduces a new algorithm for Lexical Knowledge Acquisition (LKA), using the metrics from Distributional Semantics described in the previous chapter. Section 5.1 starts with an introduction. Next, Section 5.2 describes in detail the problem that is addressed, and Section 5.3 details the settings in which the system will be trained and evaluated. Sections 5.4 and 5.5 describe the approach followed, and Section 5.6 explains the results obtained. The chapter ends up with a discussion, in Section 5.7.

5.1 Introduction

In any particular field of knowledge there is a particular set of *terms* that describe domain-specific concepts, and which do not always appear in general-purpose dictionaries, as they are not used in common life. Vivaldi et al. [2001] defines terms as "lexical units that designate concepts in a thematically constrained domain". Most scientific disciplines are continuously creating and abandoning terms as they evolve. Considering the automatic generation of hypermedia sites, this domain-specific terminology has to be taken into account by the author of the domain-specific site, for the following reason: if those terms are important, there will be sections or pages in the site that describe them.

This evolving *domain vocabularies* were traditionally captured by terminologists, but it is a task that requires a large amount of work, and which is difficult to keep up to date. Therefore, there have appeared several different procedures for automatically finding terms in technical texts. Some of them use linguistic information, such as the TERMS system [Justeson and Katz, 1995], which uses regular expressions based on the part-of-speech of the words found in the texts. On the other hand, statistical approaches usually use word and collocation frequencies, both in the domain-specific documents and in general text, in order to find which words appear more frequently in the restricted texts, and which collocations are stronger. Finally, some approaches [Vivaldi et al., 2001] use a combination of the previous ones. One of the most comprehensive reviews of Term Extraction (TE) procedures was provided by Cabré et al. [2001], who compared more than ten different approaches, and classified them according to the following characteristics: linguistic resources, strategies for term delimitation, strategies for term filtering, classification of recognised terms and results obtained.

In general, the difficulties that these procedures face are the following:

- Terms do not differ from general words either in structure or in syntactic behaviour.
- A polysemous word may be considered, with some meanings, as a term; and with other meanings as a general language word.
- Technical terms are sometimes denoted by multiword expressions, such as *relative entropy* or *conditional probability*.
- Different terms may refer to the same domain concept. This is the case of acronyms (e.g. *Lexical Knowledge Base* and *LKB*), which may also appear in plural form (*LKBs*); dialectal differences (e.g. *puntero* and *apuntador*[1]); abbreviations (e.g. *synonym set* and *synset*); and other linguistic phenomena.

For the purposes of this work, however, it is not enough to identify the terms; it is necessary to infer, to some point, their meaning, for which different strategies are followed. This work focuses in exploring new procedures for classifying the extracted terms inside semantic dictionaries, rather than reviewing the well-studied field of Term Extraction.

What follows is a new procedure that has been developed in order to semantically classify new terms found in the domain-specific texts inside the WordNet ontology. The aim is to find which unknown terms refer to people, artifacts, locations, animals, etc., with as much precision as possible in the classification of the unknown terms.

5.2 General Named Entity Identification

In Information Extraction, a *Named Entity* (NE) is an object of interest for solving a particular problem. For example, a summarisation system of political newswire articles might be interested in identifying people, international organisations, locations and dates; or a system to analyse Chemistry textbooks might be interested in finding chemical compounds and instrumental. *NE identification* (also called *NE recognition*) consists in locating and classifying the entities of interest found in a text.

In the Seventh Message Understanding Conference [MUC7, 1998], there were seven categories of entities: person, organisation, location, date, time, money and percent. The competing systems had to identify instances of these categories in some domain specific texts, such as Wall Street Journal articles.

Named Entity identification can be divided into two steps: identification of all the possible named entities (e.g. proper names); and classification in the provided categories (people, organisations, locations, etc.). Therefore, we have a flat set of seven categories and the task basically consists in finding instances of these categories in the text.

This task can be further complicated, if the systems are required to classify the entities, after this first categorisation, in sub-categories. For example, organisations can be divided into businesses, governmental institutions and international organisations.

The system that marked best at MUC-7 is the one described by Mikheev et al. [1998]. It attained an accuracy of 93.39%, which is near to the 96.95% attained by the worst human annotator. However, that

[1]Different translations used to translate the English *pointer* (in the field of Programming Languages). The first one is mainly used in Spain, and the second one in Latin America

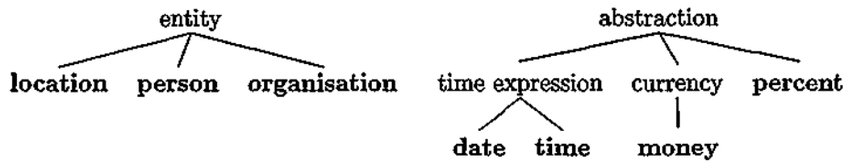


Figure 5.1: Possible taxonomies for the classes in the MUC-7 Named Entity task.

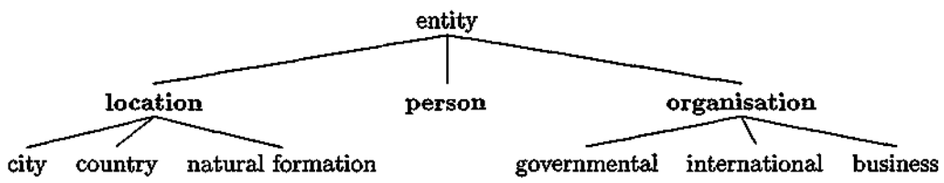


Figure 5.2: Possible extension of some entities in the MUC-7 NE task with subclasses, using the ontological approach.

system had been developed completely *ad hoc*, and since then much effort has been devoted to developing automatic methods (supervised and unsupervised) for Named Entity recognition [Borthwick et al., 1998, Bikel et al., 1999, Petasis et al., 2001, Huttunen et al., 2002, Craven et al., 1999].

5.2.1 From NE to GNE

In NE recognition, the classes of the objects of interest can be considered concepts in an ontology. For example, the seven entity types used in the MUC-7 conference could be arranged in the taxonomies in Figure 5.1. This approach seems natural for extending the task with subclasses, as some modern Information Extraction systems can perform now. Figure 5.2 shows how the *location* and the *organisation* classes could be further divided into a few subclasses.

For illustration, the following text could be the output of a Named Entity identification system with this definition. Note that the two companies cited in the text have been labelled as **business**, not as companies, because that label is more informative about those entities.

> [$_{person}$ Pierre Vinken], 61 years old, will join the board as a nonexecutive director [$_{date}$ Nov. 29]. [$_{person}$ Mr. Vinken] is chairman of [$_{business}$ Elsevier N.V.], the Dutch publishing group. [$_{person}$ Rudolph Agnew], 55 years old and former chairman of [$_{business}$ Consolidated Gold Fields PLC], was named a nonexecutive director of this British industrial conglomerate.

However, it is impossible to create an ontology containing all possible concepts. Firstly, languages are alive and under endless change, and new open-class words are created all the time. Secondly, there are many words that are particular to very specific domains, and it would be impractical to create an ontology containing all these words. More importantly, an ontology containing every possible word from every domain can even become impractical, because many of the words in it will probably never be used by most applications.

Therefore, for identifying entities from domain-specific texts, we may find that we would like to recognise instances of a concept that is not in the ontology. It would be desirable for a NE identification system to

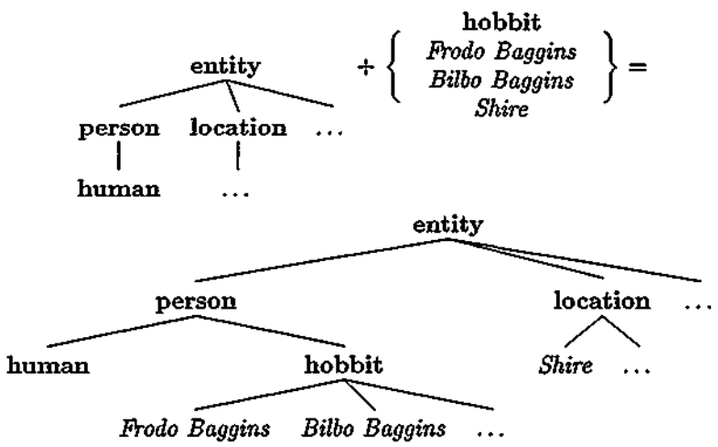


Figure 5.3: An initial taxonomy, a set of new concepts and instances, and the taxonomy extended with the concepts and instances from the set.

place the new concepts in the ontology before classifying their instances. For example, let us consider the following text taken from the foreword of *The Lord of the Rings* [Tolkien, 1968]. In this text, we can find the concept *hobbit* that does not appear in general ontologies such as WordNet, together with several of its instances, and several locations. Figure 5.3 shows how the MUC-7 ontology could be extended to contain these new concepts and instances.

> The houses and the holes of [$_{location}$ Shire]-[$_{person}$ hobbits] were often large, and inhabited by large families. ([$_{hobbit}$ Bilbo] and [$_{hobbit}$ Frodo Baggins] were as bachelors very exceptional, as they were also in many other ways, such as their friendship with the Elves.) Sometimes, as in the case of the [$_{hobbit}$ Tooks] of [$_{location}$ Great Smials], or the [$_{hobbit}$ Brandybucks] of [$_{location}$ Brandy Hall], many generations of relatives lived in (comparative) peace together in one ancestral and many-tunnelled mansion.

We can now define General Named Entity identification as a new task, in order to include the new additions. In this extended task, the immediate hyperonym of an instance or concept might be a concept that was itself learnt from the same text, such as the case with *Frodo Baggins* and *hobbit*. The concept *hobbit* should be learnt from the text and placed in the ontology, and only afterwards the other four terms can be classified as instances hobbits.

> **General Named Entity (GNE) Identification** consists in finding, for every unknown *concept or instance* found in a text, its immediate hyperonyms in an ontology.

Let us suppose that we have a conceptual ontology $\mathcal{W} = (\mathcal{L}, \mathcal{S}, f_{\mathcal{L}}, h_{\mathcal{S}}, \mathcal{R})$ where

- $\mathcal{L}$ is a set of lexical entries (words).
- $\mathcal{S}$ is a set of synsets.
- $f_{\mathcal{L}} : \mathcal{L} \rightarrow \mathcal{S}^+$ is a function that links the lexical entries with the synsets that contain them.
- $h_{\mathcal{S}} : \mathcal{S} \rightarrow \mathcal{S}^*$, *hyperonymy*, arranges the concepts and instances in a hierarchy.
- $\mathcal{R}$ is a set with other relationships.

and that we have a set of domain-specific documents $\mathcal{D}$, containing some unknown concepts and instances $\mathcal{U} = \{u_1, u_2, ..., u_n\}$.

> **GNE Identification** is the task that consists in finding, for every unknown concept or instance u_j found in a text, its maximally specific generalisations $\mathcal{G}_j = \{g_{j1}, g_{j2}, ..., g_{jn}\}$.

5.2.2 Overlapping with word-sense disambiguation

Word-Sense Disambiguation (WSD) consists in finding the meaning (or definition) with which a given word is used, which is distinguishable from other meanings potentially attributable to that word [Ide and Véronis, 1998]. Therefore, this task involves two steps:

1. Identifying all the possible meanings that a word can carry.
2. Finding, for each word, the meaning that it is actually carrying in context.

WordNet has been extensively used as a resource for WSD [Resnik, 1995, Agirre and Rigau, 1996, Kilgarriff, 1997, Wilks, 2001]. In the search for a gold standard that can be used to make comparisons between word-sense disambiguating systems, and for competitions such as SENSEVAL, [Resnik and Yarowsky, 1997] and [Kilgarriff, 1997] have proposed the use of WordNet senses to form the basis for a sense inventory. In this case, WSD can be defined as the problem of associating each occurrence of every word with the synset that contains it and whose meaning is the one with which it is being used.

Compared to the problem of finding the correct hyperonym of a subconcept or an instance, this is a different, although very related problem. In both cases, that task consists in finding the WordNet synset whose meaning is the one that is mostly related to a word in a text. Therefore, it can be expected that if a procedure produced good results for WSD, it will also produce good results for GNE if correctly adapted to the new problem. Most of the procedures for GNE explained inside this chapter had been applied to WSD before.

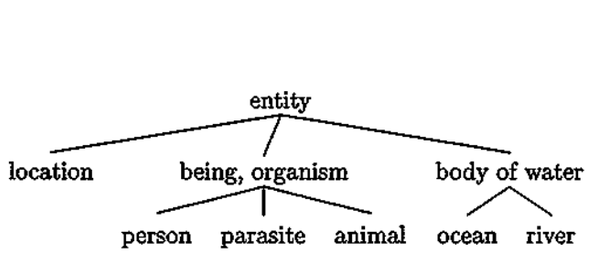
In summary, the main difference between both tasks is that in word sense disambiguation, the set of candidate synsets among which we have to choose the most similar one is reduced to the ones that contain the word, while in GNE that set is the whole ontology.

5.3 Framework for evaluation

Existing work about automatically extending ontologies with domain-specific texts tend to use different training and test data, ontologies and evaluation metrics (see Table 3.5 in Section 3.4.2). To properly compare ontology learning algorithms, we need to fix previously the training and test data, and a suitable evaluation metric.

5.3.1 Training data

In general, an Ontology Refining algorithm will need two resources: an existing ontology which has to be refined, and a text collection from which to learn the new concepts. The text collection can be used either by automatic procedures or to test hand-crafted methods to train the system. For example, Hearst [1992] used as training data the texts where she looked for co-occurring pairs of hyperonyms and hyponyms, in order to find the word patterns. In the approach that we shall follow, the training data will be used to generate, for



u_j = lice
g_j = {parasite, animal}
c_i = lice
h_{i1} = {parasite, animal}
h_{i2} = {parasite, *person}
h_{i3} = {*location, *body of water}

Figure 5.4: Example of taxonomy, an unknown relevant concept u_j, its correct generalisations g_j and the generalisations proposed by three hypothetical algorithms h_{ik}.

every concept in the ontology, the set of context words that can appear in its neighbourhood. Those sets of context words will be compared to the context of new concepts in order to decide how to classify and introduce them into the ontology.

The **existing ontology** chosen is WordNet 1.7, because there is no consensus in the existing literature, and WordNet is one of the most widely used. It is important to note that, from all the information available in WordNet, only the hierarchy of nouns which is rooted at the concept *entity* will be used. This hierarchy contains physical entities (e.g. people, locations, artifacts, animals, etc.); while the rest of the nouns refer to concepts such as *acts*, *events*, *psychological features*, etc. Most of the unknown terms found in the kind of texts that we are going to process should be classified as entities.

In order to train the algorithm, ideally, the text collection needs to be fixed so different algorithms can be compared objectively, but given the vastness of the Internet it is plausible that fixing the document bank may not be that essential, if search engines are used to find relevant documents on Internet.

The new concepts will be learnt from domain-specific texts. Three different test corpora have been built for this task: one from *The Lord of the Rings* [Tolkien, 1968], other from *The Iliad* [Homer], and one from *The Wall Street Journal* corpus. From the three of them, most of the experiments have been performed with the first one, because it was the one that contained the larger number of high-frequency unknown terms, but all of them are available and can be obtained through the web page http://www.ii.uam.es/~ealfon.

5.3.2 Evaluation metrics

Let us suppose that we have a set of unknown concepts that appear in the test set and are relevant for an specific domain: $\mathcal{U} = \{u_1, u_2, ..., u_n\}$. A human annotator has specified, for each unknown concept u_j, its maximally specific generalisations from the ontology: $\mathcal{G}_j = \{g_{j,1}, ..., g_{j,m_j}\}$. Not let us imagine that an algorithm decided that the unknown concepts that are relevant are $\mathcal{C} = \{c_1, c_2, ..., c_l\}$. For each c_i, the automatic classifier has to provide a list of maximally specific generalisations from the ontology: $\mathcal{H}_i = \{h_{i,1}, h_{i,2}, ..., h_{i,p_i}\}$.

For illustration, let us consider the ontology in Figure 5.4. If the word *lice*, appearing in some domain-specific texts, is relevant enough as to be included in the ontology, a human annotator will label it as u_j and will find a set with its maximally specific generalisations: those in g_j. Let us suppose that three different automatic classifiers have also decided that it is a relevant concept, have annotated it as c_i and have chosen as generalisations the sets h_{i1}, h_{i2} and h_{i3}, respectively. We need evaluation metrics that show that the first algorithm is eventually better than the second, which is itself better than the third one.

The following metrics have been taken, with small modifications, from Hastings [1994].

Accuracy calculates the percentage of the proposed hyperonyms that are correct:

$$Accuracy = \frac{size(Correct\ hyperonyms)}{\sum_i |\mathcal{H}_i|} \tag{5.1}$$

Parsimony is the percentage of concepts for which the set of correct generalisations is equal to the set of suggested generalisations:

$$Parsimony = \frac{|\{u_i : u_i = c_j \wedge \mathcal{H}_i = \mathcal{G}_j\}|}{|\mathcal{U}|} \tag{5.2}$$

Recall is a weaker measure than parsimony. It measures, from the relevant domain-specific concepts ($\mathcal{U}$), the percentage that were correctly identified as relevant and next correctly classified. It is considered that a concept was correctly classified if at least one of its hyperonyms was found.

$$Recall = \frac{|\{u_i : u_i = c_j \wedge \exists g \epsilon \mathcal{G}_j \text{such that } g \epsilon \mathcal{H}_i\}|}{|\mathcal{U}|} \tag{5.3}$$

Precision measures, from the chosen concepts, the percentage that were correctly classified in the ontology:

$$Precision = \frac{|\{c_j : c_j = u_i \wedge \mathcal{G}_j = \mathcal{H}_i\}|}{|\mathcal{C}|} \tag{5.4}$$

Production is the mean number of hypothesis generated for each unknown concept.

$$Production = \frac{1}{|\mathcal{C}|} \sum_i |\mathcal{H}_i| \tag{5.5}$$

While the first four metrics have to be as high as possible, production is more a descriptive metric. As the other metrics approach 1, production will approach the mean number of hyperonyms that the human annotator chose for each domain-specific concept, $\frac{1}{|\mathcal{U}|} \sum_i |\mathcal{G}_j|$.

5.3.3 Distance-based evaluation metrics

When $|\mathcal{H}_i| = |\mathcal{G}_i| = 1$, it is possible to calculate how large the distance is, in the ontology, between the proposed hyperonym and the correct one, using the metric called **Learning Accuracy** [LA, Hahn and Schnattinger, 1998]. Let us suppose that the target answer for classifying the unknown concept u_i is s_i, and the system returns the concept f_i. Let us call c_i the lowest concept that is a hyperonym of both s_i and f_i. If we call CP_i, SP_i and FP_i the lengths of the shortest paths from the top of the hierarchy to c_i, s_i and f_i, respectively; and DP_i the distance between c_i and f_i, then the LA for u_i is

$$LA_i = \begin{cases} \frac{CP_i}{SP_i} & \text{if } f_i = c_i \\ 1 & \text{if } f_i = s_i \\ \frac{CP_i}{FP_i + DP_i} & \text{otherwise} \end{cases} \tag{5.6}$$

The overall learning accuracy is the mean of the computed values:

$$LA = \sum_{i \epsilon \{1 \ldots n\}} \frac{LA_i}{n} \tag{5.7}$$

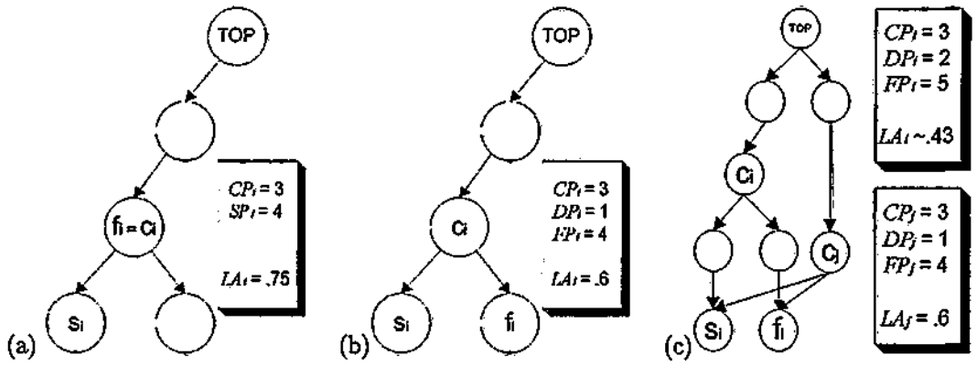


Figure 5.5: Learning accuracy in three different cases. (a) When the proposed concept is correct, but too general. (b) When the proposed concept is incorrect. (c) When there are different ways to compute Learning Accuracy.

Figure 5.5 (a) and (b) show the value of the learning accuracy in two different cases. If the output is correct, Learning Accuracy will have a value of 1. Because WordNet is not a tree, i.e. a synset can have more than one hyperonym, it may be the case that there are several ways to calculate Learning Accuracy, such as that in Figure 5.5 (c). We can redefine LA as the maximum of all of them, which corresponds to the shortest path between s_i and f_i. Therefore, LA in the example displayed would be 0.6.

However, Learning Accuracy does not take into account that the conceptual distance between a parent node and a child node in an ontology is not constant. For example, in WordNet we find that *entity* is the parent of *location*; and *womaniser* is the parent of *Don Juan*. It is evident that the distances expressed by these relationships are different, as the last two concepts are much more related to each other.

Using the studies from Resnik [1993], we can calculate the Information Content for a concept s in an ontology as the negative log likelihood $-log\ p(s)$, as described in Section 4.2.2. Using the procedure from that Section, every synset of WordNet has been automatically annotated with its Information Content.

With that information, the similarity of the two concepts s_i and f_i can be defined as the Information Content they share, i.e. the maximum of the Information Contents of the common generalisations c_i. We can calculate the following metrics:

- $ICC_i = IC(c_i)$ represents the amount of Information Content that was correctly found.
- $ICS_i = IC(s_i) - IC(c_i)$ is the amount of Information Content that was not found.
- $ICF_i = IC(f_i) - IC(c_i)$ is the amount of Information Content that was erroneously guessed.

The aim is to maximise ICC_i and to minimise both ICS_i and ICF_i. Therefore, an algorithm for GNE has to maximise the following function:

$$ICC_i - ICS_i - ICF_i \tag{5.8}$$

For example, if we have the ontology in Figure 5.6, and the concepts appear in the ontology with the frequencies shown in Table 5.1, then the Information Content for each concept is the one shown in the figure. Therefore, if we are classifying the new concept *lice*, which should be classified under *animal*, the value of the metrics based on Information Content for several possible outcomes of the classifier is shown in Table 5.2.

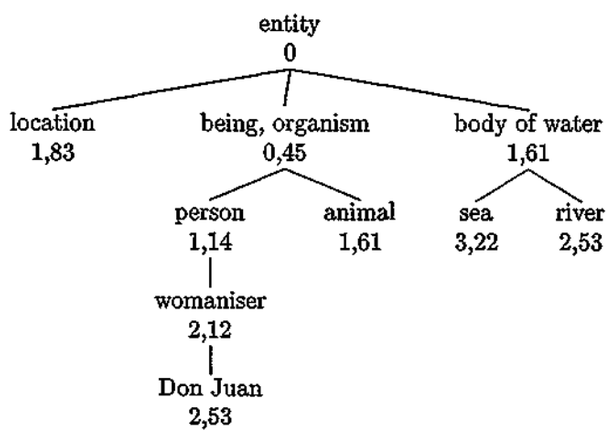


Figure 5.6: Example of taxonomy in which each node is labelled with its Information Content.

Concept	Freq.	Acc. Freq.	I.C.
entity	0	25	0
location	4	4	1.83
being	3	16	0.45
water	2	5	1.61
person	5	8	1.14
animal	5	5	1.61
womaniser	1	3	2.12
Don Juan	2	2	2.53
sea	1	1	3.22
river	2	2	2.53

Table 5.1: The concepts in the taxonomy, a hypothetical frequency for each concept, the results of adding up the frequencies of a concept's children, and the Information Content for every concept..

g_j	ICC_j	ICS_j	ICF_j
animal	1.61	0	0
*womaniser	0.45	1.16	1.67
*Don Juan	0.45	1.16	2.08
*location	0	1.61	1.83

Table 5.2: Possible generalisations suggested by a classifier, and the values of the three metrics that take into account the Information Content of each node in the ontology.

5.3.4 Test data

The ideal properties that the test data must satisfy are the following:

- It must be domain-specific.
- It must contain concepts and instances not present in WordNet, so they can be learnt.

Three collections of texts have been annotated for the GNE task: a mythological text, Homer's *The Iliad*; and a fantasy novel, Tolkien's *The Lord of the Rings*; and a portion of the *Wall Street Journal corpus* from the Penn Treebank [Marcus et al., 1993], which contains a collection of financial articles from the economics domain.

This last corpus has been used as benchmark for many other tasks in Natural Language Processing. However, the approaches described for GNE in the text make use of Distributional Semantics tools, which means that they need a representative set of contexts for any of the unknown terms in order to classify them inside a lexical ontology. Because this last corpus contains small articles about different topics, most of the domain-specific terms found in it occurred only a few times, and there was not enough contextual evidence to apply Distributional Semantic procedures; therefore it was not used for the evaluation. However, it was interesting to annotate it in order to have it available for the future, in case new algorithms are devised that do not need a big contextual evidence about the unknown terms.

The three corpora were pre-processed with the following tools:

- A tokeniser and a sentence-splitter written with regular expressions, in flex.
- The TnT part-of-speech tagger [Brants, 2000].
- A stemmer written in flex.
- Three chunkers written in C++ and Java; the first one detects Complex Quantifiers; the second one detects base Noun Phrases, and the third one detects complex verbs [Alfonseca, 2000]. All of them use transformation lists [Ramshaw and Marcus, 1995].
- A module that resolves the meaning of opening and closing quotes: whether they are genitive case markers or punctuation marks; and guessing, when a opening quote is found without a closing quote or vice versa, where the quotation ends.
- A subject-verb and verb-object detector, written in Java *ad hoc*.

After all the above-mentioned steps, all the common nouns that were not in WordNet were automatically extracted from the corpora, together with all the sequences of proper nouns. The algorithm for locating

Synset id	Words	Hyperonyms
n.wsj.00000033	James A. Talcott, Dr. Talcott	man, researcher, oncologist
n.wsj.00000124	National Cancer Institute	institute, hospital
n.wsj.00000369	Harvard University	*already in WordNet*
n.wsj.00000382	Boston University	university

Table 5.3: Unknown synsets that appear in the sentence from Figure 5.7, and the WN hyperonyms proposed for each.

Corpus	Size	Concepts annotated	No. of concepts
Lord of the Rings	540,000	freq $\geq$ 50	46
Iliad	170,000	freq $\geq$ 50	26
WSJ	12,000	all concepts	355

Table 5.4: Corpora that have been annotated, so they can be used as test set for GNE Identification procedures. The columns list the size of the corpus (number of words), the criterion for choosing the terms (all unknown terms, or only those with appear a certain number of times), and the number of concepts that were selected for each corpus.

them is described below in Section 5.4.1. From *The Iliad* and *The Lord of the Rings* only the terms with a frequency higher or equal to 50 have been marked, and their correct hyperonyms have been annotated by hand. With the algorithm described in next section, their contexts will be examined, and Distributional Semantics techniques will be used to classify them in WordNet.

On the other hand, because there was no unknown concept with as high a frequency in the Wall Street Journal (WSJ) corpus, every unknown term from this text has been annotated, regardless of its frequency.

The manual classification of the unknown terms consisted in classifying them in some of the following classes:

- A known word with a spelling mistake.
- A previously unknown word. In this case, the author identified the WordNet concepts that can be considered its maximally specific generalisations.
- A proper name already in WordNet. In this case, the new concept was annotated with the identifier of the WordNet synset.

Figure 5.7 shows an sample sentence from the corpus, and the automatic annotation that it received. All the linguistic processing was done automatically, and the annotations on how to classify the unknown concepts was done manually. This annotation includes the coreference of the unknown concepts and the proposed generalisations from WordNet. As can be seen, the automatic parser sometimes fails when parsing conjunctions and when deciding PP-attachment. There are four concepts marked in the sentence, and their manual annotation is provided in Table 5.3.

Table 5.4 shows an overview of the three corpora annotated.

5.4 An algorithm for classifying unknown terms

In order to find the hyperonyms of the unknown terms, inside WordNet, a deterministic top-down algorithm has been devised that, for each unknown term, provides a single candidate that will be considered its

```
<s id="396">
    <np det="none" person="3" number="singular" id="397" synsetId="n.wsj.00000033">
        <w c="w" abbreviation="yes" pos="NNP" stem="Dr" id="398">Dr.</w>
        <w c="w" pos="NNP" stem="Talcott" head="yes" id="399">Talcott</w>
    </np>
    <vbar time="past" tense="finite" id="400" subject="397" head="yes" args="+19947">
        <w c="w" pos="VBD" stem="lead" lexhead="yes" head="yes" id="401">led</w>
    </vbar>
    <np id="19947" conjunction="yes">
        <np id="19945" conjunction="yes" head="yes">
            <np det="indefinite" person="3" number="singular" id="402" head="yes">
                <w c="w" pos="DT" id="403">a</w>
                <w c="w" pos="NN" stem="team" head="yes" id="404">team</w>
            </np>
            <pp id="19939">
                <w c="w" pos="IN" id="405" head="yes">of</w>
                <np det="none" person="3" number="plural" id="406">
                    <w c="w" pos="NNS" stem="researcher" head="yes" id="407">researchers</w>
                </np>
            </pp>
            <pp id="19941">
                <w c="w" pos="IN" id="408" head="yes">from</w>
                <np det="definite" person="3" number="singular" id="409">
                    <w c="w" pos="DT" id="410">the</w>
                    <np id="22124" synsetId="n.wsj.00000124">
                        <w c="w" pos="NNP" stem="National" id="411">National</w>
                        <w c="w" pos="NNP" stem="Cancer" id="412">Cancer</w>
                        <w c="w" pos="NNP" stem="Institute" head="yes" id="413">Institute</w>
                    </np>
                </np>
            </pp>
            <w c="w" pos="CC" id="414">and</w>
            <np det="definite" person="3" number="plural" id="415" head="yes">
                <w c="w" pos="DT" id="416">the</w>
                <w c="w" pos="JJ" id="417">medical</w>
                <w c="w" pos="NNS" stem="school" head="yes" id="418">schools</w>
            </np>
            <pp id="19943">
                <w c="w" pos="IN" id="419" head="yes">of</w>
                <np det="none" person="3" number="singular" id="420" synsetId="n.wsj.00000369">
                    <w c="w" pos="NNP" stem="Harvard" id="421">Harvard</w>
                    <w c="w" pos="NNP" stem="University" head="yes" id="422">University</w>
                </np>
            </pp>
        </np>
        <w c="w" pos="CC" id="423">and</w>
        <np det="none" person="3" number="singular" id="424" head="yes" synsetId="n.wsj.00000382">
            <w c="w" pos="NNP" stem="Boston" id="425">Boston</w>
            <w c="w" pos="NNP" stem="University" head="yes" id="426">University</w>
        </np>
    </np>
</s>
```

Figure 5.7: Example of sentence annotated.

immediate hyperonym. The following are the initial assumptions taken before designing the algorithm:

Hypothesis 1. None of the unknown terms shall be classified in between two existing WordNet synsets. For example, no term can be classified as a hyponym of *entity* and a hyperonym of *location*, because *location* is already a hyponym of *entity* in WordNet.

There is nothing preventing a text from having some unknown term that should be placed in between two of the existing WordNet synsets. However, WordNet is very complete and it can be expected that it contains all the relevant general terms.

Hypothesis 2. One of the unknown terms can be a hyperonym of other unknown term.

For example, *The Voyages of the Beagle* contains the term *Fuegian*, which is not present in WordNet, and which is used to refer to the natives of *Tierra del Fuego*. It also talks about people such as *York Minster*, which were born at Tierra del Fuego and should be classified, in WordNet, as *Fuegian*.

This fact, combined with hypothesis 1, means that the general concepts must be inserted into WordNet before their hyponyms, and these general concepts must be taken into account as candidate hyperonyms when their hyponyms are classified in the ontology. In summary,

- Every time a concept is inserted into WordNet, automatically it will be considered as a possible hyperonym for the remaining concepts.
- Concepts that are more general must be introduced before more specific concepts (as a consequence of Hypothesis 1).

Hypothesis 3. Several unknown terms may refer to the same domain-specific concept.

This implies that, before classifying unknown terms, a coreference module must decide whether some of them refer to the same concept, and consider them all as the same. For example, the unknown term *Mr. Baggins*, from *The Lord of the Rings*, should be coreferred either with *Frodo Baggins* or with *Bilbo Baggins*, because it always refers to one of those two.

Hypothesis 4. An unknown term, in the domain-specific texts, has only one meaning throughout the text.

This simplification was taken for two reasons: firstly, in none of the test corpora processed there was a single unknown term with more than one meaning. This does not mean that there never will be a text with a domain-specific term that is not polysemous (we shall see later in Chapter 9 that it does happen sometimes), but it has been left as a minor error. Secondly, this problem is more the task of Term Identification systems that of Term Classification algorithms. The Term Identification should discover whether a word is being used in a domain-specific sense, and distinguish when it is being used with several, and next the Term Classification would classify the different senses separately.

Hypothesis 5. The unknown terms might represent concepts already in WordNet.

Although WordNet is a general lexicon, it also contains many words that can be considered as specific terms for particular domains, such as chemical compounds, locations, and scientific names for animals. Therefore, if a term is identified that belongs to WordNet, it must be coreferred with the WordNet synset.

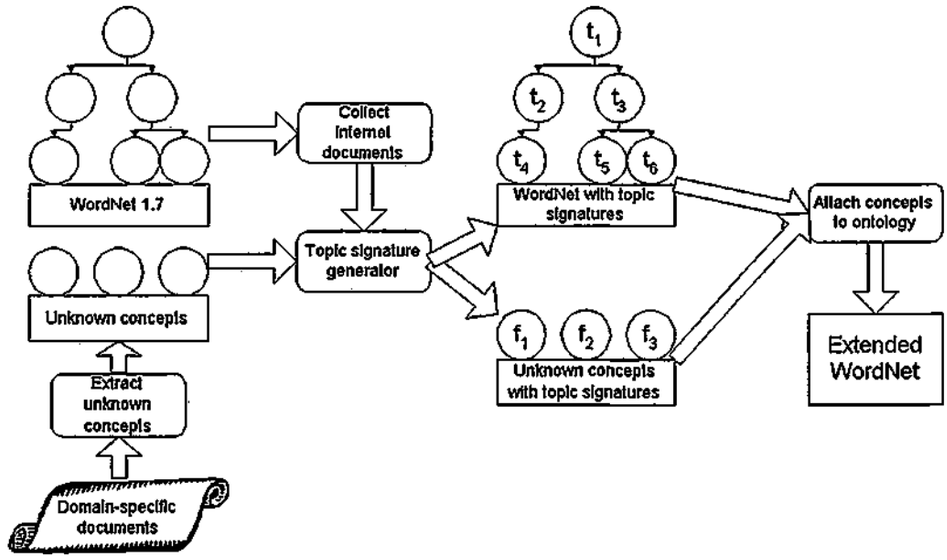


Figure 5.8: General architecture of the system that classifies unknown terms in a lexical ontology.

Hypothesis 6. For every unknown term there is one and only one correct hyperonym in WordNet.

This last hypothesis simplifies the task, because we can use a deterministic procedure to find just one synset from WordNet which is considered the most similar to the unknown concept. Note that if we use this hypothesis, and the set of found terms was the correct one (if $\mathcal{U} = \mathcal{C}$), then the four metrics described above (accuracy, parsimony, recall and precision) all have the same value. In Section 10.4.1 (*Term Classification*) possible ways to obtain more than one hyperonym for each unknown term are described.

The architecture of the new system is represented in Figure 5.8. There are three main steps:

- The identification of unknown terms (concepts and instances), which is done in the box labelled A in the figure.
- The collection of information about the synsets and the unknown concepts, to be able to calculate a similarity metric between the contexts of the synsets (boxes B and C);
- The placement of the new concepts in the ontology (box D).

The following sections describe in detail each of the system components.

5.4.1 Identification of terms

As stated in Section 5.1, Term Identification is a wholly different task of Term Classification, for which different problems have to be explored, such as to decide whether sequences of words appear together by chance of because they refer to an unknown domain-specific term; whether an unknown term is polysemous or not; and whether a previously known term is being used with a different, domain-specific meaning that was not known before, and can be the topic of complete Ph.D. theses [Vivaldi, 2002]. The Term Extraction

CollectTerms

1. Collect all the non-capitalised words that do not appear in WordNet.
2. Read a list of common title names and adjectives from a file.
3. Collect all the sequences of capitalised words. For each one of them:
   - 3.1. Check whether it overlaps with other sequence seen before.
   - 3.2. If if overlaps with one or more,
     - 3.2.1. Take the term which overlaps with it and which was seen more recently in the source files
     - 3.2.2. If they are identical, they are considered the same term.
     - 3.2.3. Otherwise, strip them off the personal titles and adjectives.
     - 3.2.4. If one is included in the other, they are considered the same term (e.g. *John Smith* and *Mr. Smith*).
   - 3.3. Else, if it is a capitalised single-word, and its stem was found non-capitalised, make them corefer (e.g. *fuegian* and *Fuegian*).
   - 3.4. Store the occurrence of the concept in the source documents in a database.

Figure 5.9: Algorithm for identifying unknown words and proper names from the source documents.

procedures used here consist of hand-crafted regular expressions rules and heuristics, but which proved good enough for the texts that have been processed. They constitute the box labelled A in Figure 5.8.

The source files are processed with the same linguistic tools with which we processed the test data (cf. Section 5.3.4): a tokeniser, a sentence-splitter, a part-of-speech tagger, a stemmer, several chunkers, and a shallow parser. Next, two different specialist modules were used to find dates and scientific names in the texts. Finally, a different module collected all unknown nouns and proper names, calculating at the same time coreferences between them. The algorithm used is displayed in Figure 5.9.

Partition of instances and concepts

The differentiation of instances and concepts at this early stage is important because, as discussed before, when we attach a new synset to the ontology its hyperonym must be already there. For instance, among the terms extracted from *The Lord of the Rings* there are several which have to be classified as hyponyms of the term *hobbit*. Therefore, the concept *hobbit* has to be in the ontology by the time we process any of those instances. It was taken as a general rule to enrich the ontology, first with the new concepts, and next with the new instances, by reordering the unknown terms before the classification.

There is discussion in literature about how to define instances and concepts. In fact, some people argue that, depending on the interpretation, anything can be considered as an instance and as a concept [Welty and Ferucci, 1999]. In our case, we decided to consider *instances* the synsets in WordNet that do not have hyponyms (*leaf synsets*) and which are not subconcepts of any of their hyperonyms. For a detailed discussion about the particular interpretation given in this work about the topic, please refer to Section B.2.2 (*Instances and concepts*).

In order to decide whether a noun refers to a concept or an instance a Maximum Entropy model has been used [Berger et al., 1996] [Ratnaparkhi, 1998]. In this framework, the problem consists in learning a probability model

$$P_{ME}(s) = \frac{1}{Z} \cdot P_0(s) \cdot \exp\left(\sum_i \lambda_i f_i(s)\right)$$

where $P_{ME}(s)$ is the probability that s is an instance, $P_0(s)$ is an initial probability distribution, Z is a

normalising constant, and f_i are binary features about the examples. Using an iterative algorithm, it is possible to obtain values for the parameters λ_i so that the model classifies the training data as best as possible. The implementation was reused from the Java package `quipu.maxent`, which is freely distributed [Baldridge et al., 2001].

Instances, in language, have some properties in common with mass nouns. For example, they are rarely preceded by the determiners *the* and *a*, or used in plural number. On the other hand, mass nouns can be quantified with weight, volume, etc. while instances cannot. These facts have been used in order to choose some features to distinguish them, such as the determiners with which terms were seen in the texts; whether they have been used in plural or not; and whether they were capitalised always, sometimes or never.

The following are several examples of features:

$f_1(s) = true$ if any word in synset s was found preceded by the determiner *the* in the documents; *false* otherwise

$f_2(s) = true$ if no word in synset s was found with any determiner in the documents; *false* otherwise

$f_3(s) = true$ if no word in synset s was found in plural form in the documents; *false* otherwise

$f_4(s) = true$ if all the words from synset s are always capitalised in the documents; *false* otherwise

As can seen from f_4, capitalisation was also used as a feature. Although not every capitalised word is an instance, many of them are, so this feature provides support in favour of considering the word an instance.

This method for distinguishing instances and concepts was tested on an extended version of WordNet that included, for each synset, information about which synsets were concepts and which were instances. This additional information has been made publicly available (http://www.ii.uam.es/~ealfon/eng/pubs.html). The overall accuracy of the algorithm, when trained on two hundred WordNet synsets, randomly chosen, and tested on other 150 synsets using a five-fold evaluation was 97.2%. Considering that about 88% of the WordNet synsets are concepts, it improves the baseline classifier that predicts that everything is a concept in nearly ten percent. Section B.2.2 describes this point in detail.

Table 5.5 lists some of the top-frequency concepts that were found in *The Lord of the Rings*, and Table 5.6 lists the concepts found in *The Iliad*. In the first case, 46 unknown words were extracted that appeared 50 or more times in the whole document, and all of them were correctly classified. The word *Gollum*, which is a character of the book, appeared written in lowercase because that character used his own name as an interjection when he spoke.

From *The Iliad* 29 unknown words were extracted, again with a frequency higher or equal to 50. The most interesting case here is the word *Ajax*, which was classified as a concept. In the text there are two characters in the text called Ajax, and the author referred to them together quite often with the phrase "`the two Ajaxes`", so it could be considered as a concept that includes both people (Figure 5.10), similar to the case of *Rama* in Figure 5.11:

With respect to the correct hyperonyms with which these extracted terms were annotated, there are a couple of considerations that have to be made:

- It has been assumed that all of these extracted terms, in the way that they are treated in the source texts used, should be placed under the WordNet synset *entity*. However, fictional characters in WordNet are placed in the hierarchy under the synset *psychological feature*. Therefore, terms such as *Jupiter* or *Frodo* should be considered mental features instead than entities.

noun	frequency	type	is_a
Frodo, Mr. Frodo, Frodo Baggins, Baggins, Mr. Baggins	1576	instance	hobbit
Sam, Sam Gangee, Gangee	1090	instance	hobbit
Gandalf, Mr. Gandalf, Master Gandalf, Gandalf Greyhame	785	instance	wizard
hobbit, Hobbit	717	concept	person
Pippin	510	instance	hobbit
Aragorn, Lord Aragorn	491	instance	man
Merry, Mr. Merry, Master Merry	374	instance	hobbit
Gollum, gollum	363	instance	hobbit
Bilbo, Mr. Bilbo, Mr. Bilbo Baggins Bilbo Baggins, Baggins, Mr. Baggins	329	instance	hobbit
orc, Orc	292	concept	person
Saruman	269	instance	wizard
Gondor	258	instance	country
Gimli, Master Gimli	248	instance	dwarf
Boromir, O Boromir, Lord Boromir	239	instance	man
Faramir, Captain Faramir, Lord Faramir	222	instance	man
Ent	147	concept	tree(?)
Bree	89	instance	town
Shadowfax	70	instance	horse
Orthanc	64	instance	tower
Anduin	59	instance	river

Table 5.5: Some of the unknown nouns identified in the domain-specific document, ordered by frequency. The third column specifies which of them are concepts and which are instances. Finally, the last column displays the synset to which the unknown synset should be attached. It is the purpose of my framework to produce that attachment.

noun	frequency	type	is_a
Trojans, trojans, O Trojans	579	concept	person
Achaeans	486	concept	person
Hector, O Hector	460	instance	man
Jove, Father Jove, King Jove	430	instance	deity, god
Achilles, O Achilles	408	instance	man
Agamemnon, King Agamemnon	187	instance	king
Ajax, Ajaxes	185	concept	man
Priam, King Priam, O Priam	180	instance	king
Patroclus, O Patroclus	156	instance	man
Minerva, Pallas Minerva	150	instance	deity, god
Apollo, King Apollo, Phoebus Apollo, Phoebus, O Apollo, O Phoebus	142	instance	deity, god
Menelaus, O Menelaus, King Menelaus	141	instance	king
Argives, Argive	133	concept	person
Atreus	129	instance	man
Peleus, King Peleus	122	instance	king

Table 5.6: Some of the unknown nouns identified in *The Iliad*.

Ajax
=> Ajax son of Telamon
=> Ajax son of Oileus

Figure 5.10: Interpretation of the word `Ajax` found in *The Iliad*. When it refers to any person called *Ajax*, then it is a concept; while when it refers to a particular person, it is an instance

.

avatar
=> Jagannath
=> Kalki
=> Krishna
=> Rama
=> Ramachandra
=> Balarama
=> Parashurama

Figure 5.11: This shows the WordNet synset *avatar* and its hyponyms. *Rama* should be an instance of *avatar*, but it is also a concept which has three different instances: the three incarnations.

On the other hand, from a Distributional Semantics points of view, they are used in the books as if they really exist with a life of their own, and hence they do behave as living entities and the contexts in which they appear reflect this fact. Men, women, animals and locations that appear in the texts do behave as men, women, animals and locations respectively, although they do not exist in the real world.

For example, all rational beings (e.g. *hobbit*, *orc* or *god*) are under the synset *person*; but they are different from the synsets *man* and *woman*.

- Some of the coreferences cannot be made automatically. In *The Lord of the Rings*, the same person is referred with three names: *Merry*, *Meriadoc* and *Mr. Brandybuck*, but there is no morphological clue to discover it. In some cases, it would be necessary a sophisticated discourse analysis to find this fact. Other example is *Gollum* and *Sméagol*, which are also two names for the same character.

- With respect to the diversity of terms, *The Lord of the Rings* contains much more types of unknown terms, such as locations, artifacts, animals, rivers, etc; compared to *The Iliad*, in which most of the high-frequency unknown concepts are people and gods.

5.5 Attachment of new synsets

Once an unknown relevant term u has been identified in the text, the next task consists in finding the place where it should be attached to the ontology. This section describes a new algorithm that performs a top-down search, and stops at the synset that is most similar to u. The search starts at the most general synset s, and compares u with it and with all of its immediate hyponyms. If s is more similar to u than any of s's children, then u is assumed to be a hyponym of s. Otherwise, we proceed downward to the most

attach(u)
u is the unknown synset,
1. Let r be the root synset in the ontology.
2. s := analyseLevel(u, r)
3. If s is an instance, return its parent; otherwise, return s.

analyseLevel(u, s)
s is the candidate synset most similar to u.
1. Get s's synset children, $\{s_1, s_2, ..., s_n\}$.
2. Calculate $d_s \leftarrow distance(u, s)$
3. For every child s_i, calculate $d_{s_i} \leftarrow distance(u, s_i)$
4. Find the concept whose semantic distance to u is the lowest
   4.1 If that concept is s, return s
   4.2 Otherwise, if that concept is s_i, return analyseLevel(u, s_i)

Figure 5.12: Pseudo-code of the algorithm for finding the correct place where the unknown synset u will be attached in the ontology

similar child found. The procedure is detailed in Figure 5.12. In Figure 5.8 above, it is represented with the box labelled D.

Note that, when the algorithm chooses the synset s in the ontology that is most similar to u, it may be the case that s is an instance, and the task definition posed the restriction that instances cannot have hyponyms. Therefore, if this happens, u will be attached to the parent node of s. For example, if the algorithm finds that the synset most similar to *Mordor* is *Australia*, then *Mordor* will be set as a child of *country*, which is *Australia*'s hyperonym, because *Australia* is an instance.

5.5.1 Signatures

In order to classify the unknown terms in the ontology, the tools that are used are based on the Distributional Semantics hypothesis. The tools used to compute the semantic distance between synsets are the topic, subject, object, and modifier signatures. The first one has already been described before, with several applications, but the others are new. We can define the signature of a concept as the list of the words that co-occur with it in certain conditions, together with their frequencies of appearance. The following types have been used:

- A *topic signature* of a concept c is the list of the words and frequencies that simply co-occur with it in the same context (e.g. in the same sentence).
- A *subject signature* of a nominal concept c is the list of verbs and frequencies for which c appears as a subject.
- An *object signature* of a nominal concept c is the list of verbs and prepositions (with frequencies) for which c appears as an argument.
- A *modifier signature* of a nominal concept c is the list of adjectives and determiners (with frequencies) that modify c inside a Noun Phrase.

The components that collect these data are represented with the boxes B and C in Figure 5.8 above. The intuition behind this procedure is that, if two words are semantically related, their signatures will also

For every WordNet synset s_i,

1. Generate a query containing all the words in s_i and its hyponyms as positive keywords, and the words in other synsets that contain the same words as negative keywords.
2. Submit the query to an Internet search engine, and collect the results.
3. Download the documents, look for the synset words in them, and calculate the frequencies of the words that occur around them, in a context of width w.
4. Store the list of words and frequencies, l_i, excluding the most common closed-class words (determiners, pronouns, conjunctions, etc).

For every list of word frequencies l_i,

1. Transform the word frequencies into weights, and produce the topic signature t_i.

Figure 5.13: Algorithm for automatically collecting lists of context words for each WordNet synset, from Internet, from Agirre et al. [2000a].

be similar. This approach has the advantage that the signatures can be automatically collected for every concept in an ontology, by collecting documents from Internet and collecting the frequencies as described by Agirre et al. [2000a] and for the unknown concepts to be classified; therefore, the classification procedure can be made unsupervised. The method is described in Figure 5.13.

The following is an example of how it would collect documents referring to the WordNet synset for *country* (sense 06621523 in WordNet 1.7). The following are the words in the synset, and the definition provided by WordNet:

> state, nation, country, land, commonwealth, res publica, body politic –(a politically organized body of people under a single government; "the state has elected a new president"; "African nations"; "students who had come to the nation's capitol"; "the country's largest manufacturer"; "an industrialized land")

The query that was produced is the following:

> "country" AND ("body politic" OR "commonwealth" OR "land" OR "nation" OR "res publica" OR "state" OR "Reich" OR "suzerain" OR "sea power" OR "great power" OR "major power" OR "power" OR "superpower" OR "world power" OR "city state" OR "ally") AND NOT ("a people"OR "area" OR "rural area")

Note that all the synset words were included, together with words which are its hyponyms. The words that appear in negative (*people*, *area* and *rural area*) are there to rule out other different senses of *country*. They were obtained by looking for other synsets containing the word *country*, which corresponded to the following meanings:

- nation, land, country, a people = (the people who live in a nation or country; "a statement that sums up the nation's mood"; "the news was announced to the nation"; "the whole country worshipped him")
- country, rural area = (an area outside of cities and towns; "his poetry celebrated the slower pace of life in the country")
- area, country = (a particular geographical region of indefinite boundary (usually serving some special purpose or distinguished by its people or culture or geography); "it was a mountainous area"; "Bible

country")

After the documents have been collected and the context words have been stored for each of the WordNet synsets, the raw frequencies are changed into weights. The reason is that some words are equally frequent in all document collections, so they do not provide contextual support and can be ruled out. Furthermore, some document collections may be bigger than others, so a normalisation is required to give the same overall weight to all signatures.

In principle, we could use any of the weight functions described in Section 4.2.1. Two weight functions have been tried, which are explained below. The two of them had been already used for word-sense disambiguation, but the χ^2 seemed to produce better results [Agirre et al., 2000a].

[Yarowsky, 1992]'s weight function is computed as follows: let us suppose that we have several lists of word frequencies $\{l_1, ..., l_n\}$, counted from document collections that contain, respectively, the words in synsets $\{s_1, ..., s_n\}$. Then, the weight for each word is given by equation 5.9.

$$\log \frac{P(w|s_i) \cdot P(s_i)}{P(w)} \tag{5.9}$$

where $P(w)$ is the overall probability of a word; $P(w|s_i)$ is the probability of w given that it is in the context of a synset s_i; and $P(s_i)$ is the probability that a word is in the context of s_i. The first two probabilities are estimated from the document collections, and the third one is assumed to be uniform.

The χ^2 function is calculated as described in section 4.2.1, repeated here for the reader's convenience. If w_j is a word, and $freq_{i,j}$ is its frequency in the frequency list l_i, then its expected mean $m_{i,j}$ is defined as

$$m_{i,j} = \frac{\sum_i freq_{i,j} \cdot \sum_j freq_{i,j}}{\sum_{i,j} freq_{i,j}} \tag{5.10}$$

The weight for synset s_j in the topic signature t_i is then

$$w_{i,j} = \begin{cases} \frac{(freq_{i,j} - m_{i,j})}{m_{i,j}}, & \text{if } freq_{i,j} > m_{i,j} \\ 0 & \text{otherwise} \end{cases} \tag{5.11}$$

For every possible comparison of synsets that can be done by the algorithm these values are calculated for every list of word frequencies, i.e. at each iteration of the top-down algorithm, we calculate the weights for the signatures of the synset s and its children s_i, between which we have to choose. The weight associated to each word is a measure of the support that a word provides that we are in the context of a certain WordNet synset.

For example, if we have to choose, at the first step of the classification, between *entity* and its children synsets, *location*, *body of water*, *being*, *thing*, etc., then for each of the synsets in that comparison we can take its list of words and frequencies, and we can use the union of all the other lists of words and frequencies as the contrast set.

Next, if *being* is chosen as the most similar synset, the next step of the comparison is between *being* and its children synsets: *person*, *animal*, *plant*, etc. Therefore, now the weights for the signature of *being* will be calculated using its children synset's context words as the contrast set, and therefore they can be different.

This procedure also requires the following two actions: first, the word frequencies should be smoothed by adding one to every frequency value, even to the words that had frequency zero in a given signature. This is done in order to overcome the sparse data problem, because it is impossible to collect every possible context

Word	Freq	w_1	w_2	Word	Freq	w_1	w_2	Word	Freq	w_1	w_2
1	1677	1.13	2.10	Dwarves	106	1.21	2.37	Doom	61	1.17	2.24
by	1124	0.48	0.61	flowers	97	1.01	1.75	yellow	61	1.14	2.15
2	658	0.91	1.49	Races	94	1.21	2.37	pink	61	1.17	2.24
killed	645	1.16	2.21	fairy	87	1.22	2.40	Barbarian	60	1.22	2.40
he	591	0.02	0.02	giant	84	1.11	2.04	Deep	58	1.20	2.35
146	307	1.17	2.24	Killed	84	1.17	2.25	Dungeons	58	1.22	2.40
145	230	1.21	2.37	Halfling	80	1.22	2.40	obtusa	57	1.22	2.40
Human	218	1.18	2.28	Cham.	76	1.22	2.40	Mixed	56	1.20	2.34
9	213	1.01	1.76	dwarves	75	1.04	1.84	Warrior	55	1.13	2.12
Elf	212	1.13	2.10	Pink	75	1.13	2.11	king	54	1.05	1.87
Gnome	150	1.19	2.31	Fairy	70	1.22	2.40	races	53	1.22	2.40
gnome	138	1.21	2.38	Cleric	61	1.22	2.40	kB.	53	1.22	2.40

Table 5.7: Some top words in the signature of the Dwarf. The second column is the frequency count, and the third column is the weight of the word, using Yarowsky's function (w_1) and Agirre's function (w_2).

in which a word can appear.

Secondly, when calculating the signature of a WordNet synset we include the sum of frequencies from the signatures of all its hyponyms, down to the leaf synsets. To illustrate this point with an example, if we were calculating the semantic distance between *causal_agency* and the new concept *hobbit*, it is probable that their contexts have few words in common. However, most of the hyponyms of *causal_agency*, such as *man*, usually appear in the same contexts in which *hobbit* does. By adding the signatures of all its hyponyms we support the decision of selecting a synset if its descendants are distributionally related to the unknown concept.

Table 5.7 shows the weight values for *dwarf*, contrasted against *fairy* and *person*, using both Agirre's and Yarowsky's weight functions. As can be seen, there is a high correlation between both weight values; however, the final classifications resulted better using the χ^2 weight function.

As an example, Table 5.8 below shows the highest frequency words in the four signatures of the concept *person*, and the weights when contrasted with its siblings and its parent synset *being*.

5.5.2 Similarity metric

Finally, in order to calculate the similarity between a synset s_i and the unknown concept u, the following computation is performed. Let t_i be the topic signature of a concept (words and weights), and l_u be the list of frequencies of co-occurring words for the unknown concept.

$$t_i = \{< w_1, w_{i1} >, ..., < w_n, w_{in}) >\}$$
$$l_u = \{< w_1, f_1 >, ..., < w_n, f_n >\}$$

where w_j is the j^{th} word in the list, w_{ij} is its weight in the topic signature t_i, and f_j is the frequency count in the contexts of u in the collection of domain-specific documents.

Next, the similarity function that has been used is the dot product of both vectors [Yarowsky, 1992]:

$$Similarity(t_i, l_u) = \sum_{j=0}^{n} w_{ij} \cdot f_j \tag{5.12}$$

Topic signature			Subject signature		
Word	**Freq**	*weight*	**Word**	**Freq**	*weight*
Rights	314	23.16	be	23	0.71
Human	162	12.89	have	14	4.24
that	161	0.00	use	10	15.09
Resources	136	19.19	write	6	20.51
Irights	109	19.94	live	5	4.59
Department	102	21.77	make	5	6.37
Chromosome	96	24.82	kill	4	24.60
information	65	11.04	work	4	24.60
Center	63	16.04	hold	3	12.65
Health	63	15.86	produce	3	5.14
not	56	0.00	suffer	3	11.29
has	56	3.75	wish	3	16.56
have	55	1.98	get	3	12.65
Object signature			**Modifier signature**		
Word	**Freq**	*weight*	**Word**	**Freq**	*weight*
of	77	0.38	innocent	16	8.28
to	54	3.15	contact	10	9.51
for	45	3.31	live	8	3.34
in	29	1.11	own	6	5.62
be	23	0.32	indigenous	6	7.28
with	15	0.82	other	5	0
on	14	1.30	same	5	6.70
from	14	2.23	controlling	5	10.25
that	11	0.00	first	3	7.57
as	10	0.06	human	3	2.76
by	9	0.35	right	3	6.36
say	6	12.45	whole	3	6.88
seek	6	13.41	particular	3	5.15

Table 5.8: Topic, subject, object and modifier signatures of the concept *person*. These are the top frequency words, together with their weights. A higher frequency does not imply that the weight will be higher, because the word may be equally frequent for every synset. Words with weight zero are more frequent in other concept's signatures.

Therefore, to find the concept that is most similar to the unknown concept u (step 4 in the algorithm) we have to find

$$\text{argmax}_i Similarity(t_i, l_u) \tag{5.13}$$

5.5.3 Combining the similarity measures

Now, if we compare the context words of an unknown term u with the topic signatures of several WordNet synset $\{s_1, s_2, ..., s_n\}$, we shall obtain a similarity between u and each of those synsets. In a similar way, we can obtain a similarity between u and those synsets by using the subject signature; and yet another two similarities for the object and the modifier signatures. In order to combine them all, the following procedure is used:

1. Normalisation: at each decision step, the similarity values provided by each signature between u and every synset s_i are divided by the overall sum, so that they add up 1. Therefore, they can be treated as probability distributions. We shall call $P_{sig_j}(s_i)$ the similarity value obtained from the signature

sig_j, normalised so all the similarity values obtained from a given signature sum to 1.

$$\sum_{i=0}^{n} P_{sig_j}(s_i) = 1 \tag{5.14}$$

2. We combine the metrics with a weighted sum, by giving a weight to each of the kind of signatures.

$$P(s_i) = \sum_{j=0}^{m} weight_j \cdot P_{sig_j}(s_i) \tag{5.15}$$

3. Finally, the weights are calculated. The baseline experiment was calculated by giving the same weight $\frac{1}{m}$ to the signatures. The final approach consisted in weighting the partial distributions P_{sig_j}, that come from each signature type, in a way that produces a weight distribution P that is equidistant to them. The distance metric chosen to compare distributions is **relative entropy**, also called **Kullback-Leibler distance**. Given two probability distributions $p(x)$ and $q(x)$, the relative entropy between p and q is defined as

$$D(p||q) = \sum_{x \epsilon \mathcal{X}} p(x) log \frac{p(x)}{q(x)} \tag{5.16}$$

Therefore, we have to calculate the weights $weight_j$ such that the final distribution P is equidistant to each weight distribution P_{sig_j}, using $D(p||q)$ as the distance metric. These weights are calculated with a simulated annealing procedure. They are initialised as $\frac{1}{m}$, and then we proceeded changing them, slowing down, until the distances $D(P_{sig_j}||P)$ all converge to the same value if possible. Finally, the synset chosen by the algorithm is

$$\text{argmax}_i P(s_i) \tag{5.17}$$

Table 5.9 shows the similarity metrics produced in the first two decisions when classifying the concept *hobbit*. The first three columns show the similarity measures by each of the three signatures, and the last column displays the overall similarity. In both cases the outcome was correct.

5.6 Experiments and Results

WordNet 1.7 contains 74,487 noun synsets, of which 41,071 are located under the synset *entity*, and are therefore relevant for this study. For this work, the topic, subject, object and modifier signatures were collected for roughly 3,900 of them, by choosing a representative sample of all the main groups of entities: locations, animals, people, artifacts and bodies of water.

In the case of *The Lord of the Rings*, a total of 46 concepts appeared 50 or more times in the texts; therefore, there was some amount of contextual evidence to be able to classify them.

Concerning the evaluation metrics, it is assumed that the set of unknown terms has been correctly identified, and that every unknown term has exactly one hyperonym in WordNet (only one hyperonym is identified for each term with the top-down algorithm). Therefore, *production* will always have a value of 1, and *precision*, *recall*, *production* and *accuracy* will always have the same value. Two different versions of **accuracy** have been studied. The first one, *strict accuracy* simply calculates the percentage of the proposed

	First decision: entity				
synset	**synset Id**	sim_1	sim_2	sim_3	*total*
being, organism	n00002908	0.51	0.46	0.29	0.4216
body of water	n07411542	0.12	0.24	0.15	0.1680
location	n00018241	0.18	0.11	0.14	0.1430
thing (object)	n00002254	0.02	0.07	0.08	0.0528
cell	n00004081	0.02	0.05	0.09	0.0504
thing (anything)	n03781420	0.04	0.02	0.08	0.0446
variable	n07599876	0.02	0.01	0.05	0.0245
(14 more)	...	...	...	...	...
	Second decision: being				
synset	**synset Id**	sim_1	sim_2	sim_3	*total*
human	n00005145	0.35	0.46	0.23	0.3453
animal	n00010787	0.30	0.31	0.22	0.2764
host	n01015823	0.04	0.03	0.10	0.0572
mutant	n08290953	0.02	0.06	0.06	0.0436
parasite	n01015154	0.03	0.02	0.07	0.0402
flora	n00011740	0.02	0.03	0.06	0.0347
mascot	n08257648	0.02	0.01	0.04	0.0216
(32 more)	...	...	...	...	...

Table 5.9: Similarity values for each of the decisions that have been taking when classifying the unknown concept *hobbit*. In the first place, when deciding between *entity* and its children, the winner synset was *being, life form*. In the second decision, when deciding between this last synset and its children, the winner was *human*. Both decisions were correct, because they were the synsets more similar to the meaning of *hobbit*.

hyperonyms that are correct:

$$Accuracy = \frac{size(Correctly\ classified)}{size(\mathcal{U})} \tag{5.18}$$

The second one, *lenient accuracy*, is the percentage of times that the system proposed a hyperonym that can be considered valid, although it is not the best one. For example, in one of the experiments, the unknown concepts *Boromir* and *King_Theoden* were classified as

> boy – (a friendly informal reference to a grown man; "he likes to play golf with the boys"),

These outputs were considered as correct when calculating the lenient accuracy, because they are grown men, although they are not correct for the *strict accuracy* metric, for which to be correct they should have been classified as *man*.

Other metrics that can be measured, because of the characteristics of the top-down algorithm, are **Correct decisions** (C.D.), the percentage of times that a correct decision was chosen at each iteration of the algorithm; and **Correct position** (C.P.), which measures, at each step in the search, when the different children synsets are ordered according to the signatures, the mean position of the correct one. Ideally, this metric has to be as low as possible.

The other metric that is used is *Learning Accuracy*, defined above in Section 5.3.3.

5.6.1 Results on weighting the signatures

Keeping constant the number of signatures at three: topic, subject and object, the two methods to combine them were tried: the baseline, using a uniform weight, and the simulated annealing, using relative entropy. Results are displayed at Table 5.10. The procedure that uses relative entropy to find the correct distribution

Method	Accuracy		L.A.	C.D.	C.P.
	strict	len.			
Uniform	13.04%	23.91%	0.34	71.09%	1.98
Entropy	13.04%	28.26%	0.38	73.44%	1.95

Table 5.10: Comparison of two methods to combine the results provided by the signatures. Columns represent: strict and lenient accuracy; Learning Accuracy; the percentage of times that the algorithm chose the correct decision (C.D); and the mean position of the correct decision to choose (M.P.)

Method	Accuracy		L.A.	C.D.	M.P.
	strict	len.			
Topic	6.52%	17.39%	0.30	68.21%	2.30
Modifiers	16.28%	21.74%	0.29	62.96%	4.47
Subject	10.87%	23.91%	0.30	68.80%	3.06
Object	17.39%	28.26%	0.38	71.43%	2.63
T+S+O	13.04%	28.26%	0.38	73.44%	1.95
TSOM	10.87%	21.74%	0.35	70.31%	2.00

Table 5.11: Results using different signatures.

produced equal or better results for all the metrics than the uniform weighting of the similarity metrics.

5.6.2 Results on different signatures

In the following experiments, different combination of the signatures were tested. Table 5.11 shows the results.

The signature that produced the worst results was the modifier signature: the learning accuracy is the smallest, and the mean position of the correct concept at each decision is very high, nearly 1.5 over the next mark. Also, when used with the others, the modifier signature greatly degrades the results. A manual examination of the modifier signatures revealed that they contained a large amount of mistakes, due to parsing errors: words that should not be considered modifiers were part of these signatures. That might be the reason why this signature was not accurate. However, on the other hand, also the subject and the object signatures contain some errors, and that does not prevent them from being useful. It is also possible that nouns' selectional preferences for adjectives are less informative for our aims than verbs' selectional preferences for nouns, exemplified by the subject and object signatures. More experiments are necessary in order to study this particular case.

The other three signatures produced acceptable results, and the best mark was attained by combining them all. Most of the errors were produced at the lowest levels of the ontology, when deciding between semantically similar synsets such as *man* and *woman*, for which the context is not much help.

5.6.3 Comparison with other approaches

Table 5.12 shows results obtained by our system, labelled T+S+O, compared to other similar systems. Note that they call *precision* what we call *accuracy*: the percentage of concepts that were correctly classified.

These have some important differences, so the results are not really comparable. First, the ontologies used in each approach are different. This can have dramatic consequences on the evaluation because, as shall be seen, WordNet has some characteristics that complicate the task. For example, geographical locations

System	Accuracy		L.A.
	strict	lenient	
T+S+O	28.26%	36.96%	0.44
	single	set	
[Hastings, 1994]	19%	41%	-
Hahn et al [1998]	21%	22%	0.67
Hahn et al [1998]-TH	26%	28%	0.73
Hahn et al [1998]-CB	31%	39%	0.76

Table 5.12: Comparison of my approach with two different systems. The lines labelled TH and CB show two improvements performed to the basic algorithm by Hahn and Schnattinger [1998]

(e.g. continents or peninsulas) in WordNet are classified as *object*, while political locations (e.g. countries or counties) are classified as *location*. However, the context of these two kinds of entities are very similar, while being very different to the context of other *objects* such as *artifacts*. This provoked that the algorithm "incorrectly" classified most geographical locations, such as mountains or woodlands as *locations*.

Secondly, the systems by both Hastings [1994] and Hahn and Schnattinger [1998] are probabilistic, and they do not return a single hyperonym but a list of possible hypothetical hyperonyms. In Table 5.12, the column labelled **single accuracy** shows the percentage of outcomes in which the systems returned a unique hyperonym and that one was correct (like the algorithm described here); and the column labelled **set accuracy** is the percentage of times that the system returned one or more outcomes among which the correct one is. This means that the metric that has to be compared with our *strict accuracy* is the *single accuracy*, and the results are promising: the algorithm described here performed better than Hastings's, and with results comparable to the best results of Hahn and Schnattinger [1998]'s. The measure of LA is much lower for the approach described here; however, in order to make a complete comparison, we should use the same lexical ontologies, because the problem is much harder with very large ontologies such as WordNet than with small, domain-specific ontologies.

5.7 Summary and discussion

This section describes a new algorithm that can be used to place high-frequency unknown terms inside lexical ontologies, using contextual evidence to guide the classification. The steps of the algorithm are the following:

- Identification of unknown terms.
- Collection of contextual data for each of the nodes in the ontology: topic, subject and object signatures.
- Collection of contextual data for the domain-specific terms.
- Classification of the unknown terms, using a top-down algorithm, and combining at each step the results given by each of the kinds of signatures.

At present, to the author's knowledge, the algorithm described here is the only fully unsupervised method to extend a lexical ontology with unknown concepts taken from domain-specific documents. In principle, it could be applied to different languages and domains as is, if the linguistic tools are available and they respect the markup style. It is highly versatile, and it allows the attachment of new concepts to any intermediate level in an ontology, not only at the leaves. It has been shown that it is able to tackle big ontologies with the size of WordNet.

The advantages with respect to the most popular cluster algorithms for building ontologies [Faure and Nédellec, 1998] are several. In the first place, the ontologies produced are not necessarily binary trees. Secondly, every single synset in the ontologies generated with this approach stems from a word, while many of the clusters in their approach do not have a counterpart in the language, so it is necessary that a human judge name it. Thirdly, this approach can be used to extend existing general ontologies, while their approach generates the ontology from scratch, so we may start with widely used ontologies such as WordNet. Finally, the approach is fully unsupervised, and the whole ontology can be obtained without any human intervention.

Two new tools called *subject signature* and *object signature* have been introduced, and they have proven, in some cases, to be more accurate than the *topic signature*, although their main drawback is that a single text provides just a few samples, and that might generate a sparse data problem. As can be seen in the sample signatures in table 5.8, most of the frequencies have very low values.

In theory, the algorithm can also be used to create a new ontology from scratch. In this case, however, we must be careful that the concepts are learnt from the most general to the most specific one, because once a concept is attached to the hierarchy it is not possible to move it from its position.

This work can also be used to test the degree of adequacy between existing ontologies, such as WordNet, and the usage of concepts in language. For instance, *fairy* and *dwarf* are considered, in WordNet, hyponyms of the concept *psychological feature*, and thence they are located far from *animate being*; but they are always used in language in the same way as animated beings, in the sense that they are usually selected by the same verbs and have similar complements.

Other previous approaches, such as the systems reported by Hahn and Schnattinger [1998] and Hastings [1994], make use of resources such as the selectional restrictions for verbs, or scripts of newswire articles. These are used for reducing the hypothesis space of the candidate hyperonyms. However, they do not indicate any way to learn these rules automatically and, if they have to be encoded by hand, it makes it difficult to be ported to different domains.

The main drawback of the proposed system, as it is now, is that the signatures need to have a certain size in order to provide reliable classifications. An unknown concept that only is cited once will produce a topic signature with as many entries as words are in the same sentence; and the subject and object signatures will have, in the best case, just one entry.

Secondly, there are some semantic distinctions that are very difficult to represent with the contexts. During the classification, the most common error was due to the fact that all the eight instances of *hobbit* were classified as *man*, with *hobbit* as the second option, because the topic signature of *man* is much more complete; also, every man and woman were classified as man, regardless of their sex, because the words that can appear in the context of men and women are very similar; and so, for example, men and women can appear as the subject of more or less the same verbs (there are only a few exceptions).

As a matter of fact, it must be noted that some attachments would not have been straightforward even for a human annotator. For example, *ent*, in *The Lord of the Rings*, represents a kind of trees that talk and walk and behave like persons, and *Treebeard* is an instance of that concept. One possible hyperonym was *plant*, but because it appears in the text as subject of movement or speech verbs, the similarity function produces nearly a draw between *person* and *animal*, with a slight advantage toward *person*, but it completely disregards *plant*.

Finally, the process of collecting the information from the Internet, with the current technology, is very slow and depends on the characteristics of the network. This is the main reason why the procedure was not applied to the complete WordNet ontology, and it has also been noted by other research which uses topic

signatures [Agirre et al., 2000a].

Next chapter describes a way to improve this algorithm with other methods in order to overcome these problems.

Chapter 6

The Documental Database

The purpose of the off-line processing in the proposed architecture is the construction of a static database, containing information that might be relevant for the users that access the texts. The different analysers that process the input texts will store their output inside XML files and a relational database, which will be used during the on-line processing. The information includes:

- The original documents, with annotations about morphological and syntactic information.
- The dates that are present in the documents, and references to the sentences where they appear.
- Other unknown terms found in the documents, such as unknown names of locations, people, or any other kind of entities, as explained in the previous chapter. These will be stored together with their classification in WordNet, and with all their occurrences in the texts.
- It is possible to add specialist modules for identifying other kinds of terminology. This possibility has been deployed and tested with a specialist module that identifies scientific names of animals, plants, and other life forms, but the framework could accept analysers trained for any other kind of entity.

This chapter describes the ways in which this information is collected from the original documents.

6.1 Linguistic annotation

All the original documents were pre-processed with some linguistic tools, already enumerated in the previous chapter (Section 5.3.4): a Flex tokeniser, a sentence splitter, the TnT part-of-speech tagger, a Flex stemmer, transformation-list Noun Phrase and Verb Phrase chunkers; a module to find the function of opening and closing quotes, and a parser programmed *ad hoc*. Finally, other module identifies the chapter and section boundaries in the text, based on regular expressions, keywords and heuristics.

Each of these receives as input the output of the previous tool, in the form of an XML document that follows a DTD designed for this architecture, but similar in many respects to the Corpus Encoding Standard [CES, Ide, 2000]. The output is written in the document as new XML entities or attributes. A module is also able to correct errors produced by the previous modules. For example, the distinction between a past-tense verb and a past participle is sometimes difficult to make by the part-of-speech tagger, but the verb phrase chunker can correct some of the errors, such as when the verb *to have* is followed by a verb annotated as a past-tense verb; in this case, this annotation is changed into a past participle.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE document SYSTEM "yorkie.dtd">
<document>
<header/>
<body id="0" nextId="25586">
<chapter id="25570">
<header id="25567">
<p id="26">
<s id="27">
<np det="none" person="3" number="singular" id="28">
<w c="w" pos="NN" stem="CHAPTER" id="29">CHAPTER</w>
<w c="w" abbreviation="yes" pos="NNP" stem="I" head="yes" id="30">I.</w>
</np>
</s>
</p>
</header>
<section id="25562">
<header id="25561">
<p id="6281">
<s id="6282">
<np det="none" person="3" number="singular" id="6283">
<w c="w" pos="NNP" stem="FERNANDO" id="6284">FERNANDO</w>
<w c="w" pos="NNP" stem="NORONHA" head="yes" id="6285">NORONHA</w>
</np>
<w c="," pos="," id="6286">,</w>
<advp entity="date" id="6287" calendarDate="123456789/123456789/123456789/20/2/1832/0/t13/">
<w c="w" pos="NNP" stem="FEBRUARY" id="6288">FEBRUARY</w>
<w c="cd" pos="CD" id="6289">20</w>
<w c="," pos="," id="6290">,</w>
<w c="cd" pos="CD" id="6291">1832</w>
</advp>
</s>
</p>
</header>
```

Figure 6.1: Sample of the start of a document that contains a chapter from *The Voyages of the Beagle*, with some annotations for the document structure, syntactic chunks and dates.

The beginning of a document that contains a chapter of *The Voyages of the Beagle* is shown on Figure 6.1. Figure 6.2 shows a paragraph taken from the same chapter. For a larger description of the linguistic tools used, please refer to Appendix B.1 (*Linguistic tools*).

6.2 Term classification with hyponymy patterns

Term classification can be performed, with some amount of errors, with the procedures described in the previous chapter; however, a simple modification of the previous algorithm can still improve the accuracy of the procedure.

If we examine the mistakes committed by the Distributional Semantics algorithm (in Section 5.5), we find that it is difficult for it to distinguish between concepts that can appear in similar contexts. For example, the topic signatures of the WordNet synsets *adult_male* and *adult_female* are very similar, and many mistakes were people that were classified in the wrong sex. Due to the same reason, when processing excerpts from

```
<p id="6292">
<s id="6293">
<w c="w" pos="IN" id="6294">As</w>
<w c="w" pos="RB" id="6295">far</w>
<pp id="21236">
<w c="w" pos="IN" id="6296" head="yes">as</w>
<np number="singular" person="1" id="6297">
<w c="w" pos="PRP" head="yes" id="6298">I</w>
</np>
</pp>
<vbar voice="passive" time="past" tense="finite" id="6299" args="+6302">
<w c="w" pos="VBD" stem="be" head="yes" id="6300">was</w>
<w c="w" pos="VBN" stem="enable" lexhead="yes" id="6301">enabled</w>
</vbar>
<vbar tense="infinitive" id="6302" subject="*6299" args="+6311">
<w c="w" pos="TO" id="6303">to</w>
<w c="w" pos="VB" stem="observe" head="yes" lexhead="yes" id="6304">observe</w>
</vbar>
<w c="," pos="," id="6305">,</w>
<pp id="23660">
<w c="w" pos="IN" id="6306" head="yes">during</w>
<np det="definite" person="3" number="plural" id="6307">
<w c="w" pos="DT" id="6308">the</w>
<w c="w" pos="JJ" id="6309">few</w>
<w c="w" pos="NNS" stem="hour" head="yes" id="6310">hours</w>
</np>
</pp>
<np number="plural" person="1" id="6311">
<w c="w" pos="PRP" head="yes" id="6312">we</w>
</np>
<vbar time="past" tense="finite" id="6313" args="+6320">
<w c="w" pos="VBD" stem="stay" lexhead="yes" head="yes" id="6314">stayed</w>
</vbar>
<pp id="23662">
<w c="w" pos="IN" id="6315" head="yes">at</w>
<np det="none" person="3" number="singular" id="6316">
<w c="w" pos="DT" id="6317">this</w>
<w c="w" pos="NN" stem="place" head="yes" id="6318">place</w>
</np>
</pp>
<w c="," pos="," id="6319">,</w>
<np det="definite" person="3" number="singular" id="6320">
<w c="w" pos="DT" id="6321">the</w>
<w c="w" pos="NN" stem="constitution" head="yes" id="6322">constitution</w>
</np>
<pp id="23664">
<w c="w" pos="IN" id="6323" head="yes">of</w>
<np det="definite" person="3" number="singular" id="6324">
<w c="w" pos="DT" id="6325">the</w>
<w c="w" pos="NN" stem="island" head="yes" id="6326">island</w>
</np>
</pp>
<vbar time="present" tense="finite" id="6327">
<w c="w" pos="VBZ" stem="be" lexhead="yes" head="yes" id="6328">is</w>
</vbar>
<w c="w" pos="JJ" id="6329">volcanic</w>
<w c="," pos="," id="6330">,</w>
<w c="w" pos="CC" id="6331">but</w>
<w c="w" pos="RB" id="6332">probably</w>
<w c="w" pos="RB" id="6333">not</w>
<pp id="23666">
<w c="w" pos="IN" id="6334" head="yes">of</w>
<np det="indefinite" person="3" number="singular" id="6335">
<w c="w" pos="DT" id="6336">a</w>
<w c="w" pos="JJ" id="6337">recent</w>
<w c="w" pos="NN" stem="date" head="yes" id="6338">date</w>
</np>
</pp>
</s>
</p>
```

Figure 6.2: Sample of a paragraph from *The Voyages of the Beagle*, with some syntactic annotation.

The Lord of the Rings [Tolkien, 1968], all hobbits were classified as men.

The approach described by Hearst [1992, 1998], and used also by Kietz et al. [2000] for an Ontology Refinement (OR) system, consists in applying regular expression patterns in order to learn new hyperonymy relations. These patterns can be either hand-crafted or automatically collected from free texts by looking for pairs of (hyperonym, hyponym) that co-occur in the same sentence, and by looking at how they are related in the sentence.

For the convenience of the reader, the example from Section 3.4.2 is repeated here. From sentence (9), taken from Hearst [1998], a system can discover that the pattern such `NPs` as `{NP,}* NP` usually states a hyperonymy relation, if *Herrick*, *Goldsmith* and *Shakespeare* appear as hyponyms of *author* in the initial ontology. That pattern can be used later to learn relationships between new concepts. We shall call these patterns *hyponymy patterns*.

(9) ...works by such authors as Herrick, Goldsmith and Shakespeare...

Kietz et al. [2000], using hand-coded patterns, and working with the German language, quantified the error rate of this procedure in 32%. As described by him, this procedure has several drawbacks:

- The list of patterns was compiled by hand.
- If a concept never appears inside one of these patterns, the system cannot classify it.
- The error rate is high, so it is necessary that a user validates the program's output.

The approach taken by Hearst [1998], by looking for hyponymy patterns and then extracting the hyponymy relationships can help improve this weak point in the previous algorithm because, when the extracted relationship is correct, it is usually relevant. However, as she notes, the hyponymy patterns used to find new hyperonymy relationships can generate a large number of mistakes, either because the extracted relation is far too general (e.g. `hyperonym(exercise, thing)`); because they are subjective opinions with little interest (e.g. `hyperonym(Gaslight, classic)`, referring to the film *Gaslight*); or because of parsing errors. On the other hand, when they are correct, they usually state relevant hyperonymy relationships that can be used to distinguish subtle conceptual differences (e.g. sex and age) that are difficult to identify with the algorithm described in the previous chapter.

Hence, the improvement proposed consists of the following:

- The use of the hyponymy patterns only as a support for the top-down classifier for making the decisions, when the topic signature gives a similar weight to several concepts.
- The automatic extraction of a different set of hyperonymy patterns for every level of the WordNet hierarchy.

6.2.1 Automatic extraction of hyponymy patterns

As [Hearst, 1998] proposes, hyponymy patterns can be extracted automatically from texts by looking at sentences that contain a pair (*hyperonym*, *hyponym*) from WordNet. After defining First Order Predicate Logic (FOPL) predicates to represent several kinds of syntactic dependencies, it is possible to extract the dependencies between the hyperonym and the hyponym that co-occur in the same sentence.

To obtain the hyponymy patterns that apply to each WordNet synset, the following steps are followed:

- For each WordNet synset, a query is automatically constructed for the Altavista Internet search engine, following the procedure detailed in [Agirre et al., 2000a], and a set of documents is collected that contain the words in that synset.
- The documents are processed with a Flex tokeniser, a sentence splitter, the TnT part-of-speech tagger, a Flex stemmer, and a transformation-list Noun Phrase chunker, as described in Section 5.3.4.
- The sentences from those documents that contain both any of the synset words and any of its hyperonym's words are selected.
- The system extracts the hyponymy patterns from them, using the FOPL predicates, and prunes the low-frequency ones.

For example, the following are some of the patterns that were extracted from the texts. The first one shows the case in which the verb *to be* functions as a copula; the second and the third phrases show appositive constructions; and the last case shows how a prepositional phrase can indicate a hyperonymy relationship.

(1) `Shakespeare was a first-class poet`
hyperonym(N2, N1) :- subject(N1, be), object(N2, be)
(2) `Shakespeare, the poet, ...`
hyperonym(N2, N1) :- appositive(N2, N1)
(3) `The English dramatist, Shakespeare, ...`
hyperonym(N2, N1) :- appositive(N1, N2)
(4) `...the city of Seville...`
hyperonym(N2, N1) :- pp_modifier(of, N1, N2).

These patterns are extracted at each level of the WordNet hierarchy, from documents downloaded from Internet corresponding to roughly 3,900 of the WordNet synsets. The experiments suggest that some rules such as (1), (2) and (3) are general and appear at every level, but rule (4) applies only in a few cases, specially for geographic regions such as *city*, *kingdom* or *valley*.

6.2.2 Modifications to the original algorithm

Firstly, for each of the unknown terms, all the patterns are used in order to find candidate hyperonyms, but only those candidates for which the pattern that extracted them applies are retained. For example, if a pattern such as

hyperonym(N2, N1) :- pp_modifier(of, N1, N2).

extracts from a text two candidate hyperonyms, *city* and *man*, because that pattern had been extracted for the WordNet synset *city*, and not for the WordNet synset *man*, only the first candidate shall be retained.

Secondly, the top-down algorithm is modified so, at each level, if one of the synsets that are candidate hypernyms has one descendant in the ontology which had been suggested by the patterns, the support for choosing that synset increases. In particular, it is multiplied by a factor which decreases with the depth of that descendant. For instance, in the classification for *hobbit* shown in table 5.9 above, if the patterns

Method	Accuracy		L.A.	C.D.	M.P.
	strict	len.			
Original	13.04%	28.26%	0.38	73.44%	1.95
O+Patterns	28.26%	36.96%	0.44	76.56%	1.85

Table 6.1: Results without and with patterns. The columns represent strict and lenient accuracy; Learning Accuracy; the percentage of times that the algorithm chose the correct decision (C.D); and the mean position of the correct decision to choose (M.P.)

had suggested that *hobbit* could be a hyponym of *animal*, its weight would have been multiplied by N, and *animal* might have been the decision taken; if they had suggested that *hobbit* could be a hyponym of *domestic animal* (which is a child of *animal*), then it would have been multiplied by $N/2$, and so on.

If the factor N is too large, then the errors of the hyponymy patterns will spoil correct classifications based on the signatures, but if they are too small then they will not affect the classification at all. The value that was finally taken was set by hand after performing several experiments, as the one that produced the best results in the overall classification.

In this way, a double objective is fulfilled:

1. The directed search of the top-down algorithm helps in that most of the erroneous hyperonyms suggested by mistakes of the patterns are never considered, because the search does not proceed near them.

2. The hyperonyms suggested by the patterns help the top-down algorithm when the decision is difficult because two concepts appear in very similar contexts, such as the male-female distinction.

6.2.3 Experiments and Results

The new version was tested with the same 46 concepts taken from *The Lord of the Rings* [Tolkien, 1968], that were classified with the previous version of the algorithm (cf. Section 5.6). For the moment, only rules for appositive constructions and the verb *to be* in a copular construction are considered, but the algorithm could easily be extended for other syntactic relationships such as for prepositional phrases (as shown in the example sentences above) or other kinds of syntactic dependencies.

The results are displayed in table 6.1. Without using hyponymy patterns, 13.04% of them were correctly attached to the ontology, although most of the incorrect classifications were due to decisions that could hardly be decided from the context words (e.g. all hobbits and women were classified as men). This improved up to 28.26% by using the patterns, i.e., the strict accuracy more than doubled.

With respect to *lenient accuracy*, again some concepts were given sensible classifications, although they were not the expected one. For example the concept *orc* was finally classified as *bozo* with the meaning "a stupid fool". Because orcs are considered stupid in the book, the classification was marked as correct for calculating the lenient accuracy, although that was not the expected classification. Using the patterns, lenient accuracy improved from 28.26% to 36.96%.

Concerning Learning Accuracy, it improved from 0.38 to 0.44; the times that the algorithm chose a correct decision improved from 73.44% to 76.56%, and the average position of the correct synset in these decisions decreased from 1.95 to 1.85. As can be observed, there was an improvement for every metric used in the evaluation.

First decision: entity			
synset	**synset Id**	*total*	*hypon*
causal agency	n00004753	0.3860	1.25
being, organism	n00002908	0.3642	1.25
location	n00018241	0.1180	
body of water	n07411542	0.0568	
thing	n03781420	0.0414	
thing	n00002254	0.0191	
entity	n00001740	0.0032	1.125
(15 more)	...	...	

Second decision: causal agency			
synset	**synset Id**	*total*	*hypon*
person	n00005145	0.9975	2.5
causal agency	n00010787	0.0025	

Third decision: person			
synset	**synset Id**	*total*	*hypon*
man	n00005145	0.3344	
woman	n00010787	0.2821	
hobbit	n.lotr.01	0.2268	5
appointee	n07716947	0.0277	
lover	n07729287	0.0236	
(295 more)	...	...	

Table 6.2: Similarity values for each of the decisions that have been taken when classifying the unknown concept *Frodo*, and factors provided by the hyponymy patterns.

Table 6.2 shows the similarity values when classifying the concept *Frodo*, taken from the same book. *Frodo* appears, in particular, in sentence (10):

(10) Mr. Frodo is as nice a young hobbit as you could wish to meet

This sentence contains one of the hyponymy patterns, and indicates that *hobbit* (which had been learnt before) might be a good candidate as a hyperonym. Therefore, the support for each synset was modified accordingly. The synsets *causal agency* and *being*, which are grandparents of *hobbit*, have their support increased by a factor of 1.25; and *entity*, which is a grand-grandparent, increases it by a factor of 1.125.

The first two decisions (*causal agency* and *person*) were not altered by the use of hyponymy patterns, because the synset whose support was multiplied by the highest factor was also previously the one with the largest support. However, in the third case, there were three synsets with a high support given by the context: *man*, *woman* and *hobbit*. Here, the hyponymy patterns helped in choosing the correct option.

6.3 Identification of time expressions and events

Contextual information and hyponymy patterns are useful when classifying unknown terms about whose meaning we do not know anything beforehand; however, date and time expressions are easily identifiable with regular expressions and they are rarely ambiguous, so it is probably best to detect them with other tools. Hence, the part of the system that discovers time expressions is completely independent from the modules for classifying terms, described above.

A story usually consists of one or more episodes, each one of which consists of a set of **events**, which may be **actions** or **states**. The text usually provides additional information about the events, such as the agent or patient, the objects involved, and the time and location settings [Bell, 1998]. Interestingly, episodes and events are not usually provided in chronological order. Brewer [1985] distinguishes between the order in which a story is told (the *discourse structure*) and the chronological ordering of the events (the *event structure*). If we want to understand the meaning of a text, it is important to be able to reconstruct the order in which events happened. Studies by linguists show that discourse structure and event structure are usually different, and this applies to different domains. For example, Bell [1998] provides several single-sentence newswire stories each of which contains up to five events, with many different reorderings.

In order to understand a text it is important to be able to reconstruct its *event structure*. An automatic procedure for doing this has potentially many useful applications, specially in multi-document summarisation [Mani and Wilson, 2000]. In the Sixth and the Seventh Message Understanding Conference [MUC6, 1995] [MUC7, 1998], the task called *Scenario Template* included identifying time expressions, and assigning a calendar time only to the scenario event types (joint ventures in MUC-6, and rocket launchings in MUC-7). However, the scores were low. The SRA system, which scored best at that task, found only around 35% of the start and end times of events; and more than half of the answers it provided for this task were incorrect.

Mani and Wilson [2000] extended the interpretation of time expressions in MUC to include expressions that are relative to the time in which the document was written, such as *today* or *three days ago*. Filatova and Hovy [2001] describe a method that looks for events in newswire articles, and assigns calendar times to them whenever possible.

A big handicap for finding the event structure of texts is the fact that yet there are not good guidelines for annotating texts with temporal information. Setzer and Gaizauskas [2001] describe a project that aims at producing a training set for this task. The annotators had a framework to help them annotate the text with temporal expressions, mark the events that they modify, and reorder these events in time. Even after the framework had prompted the annotators to specify the temporal relationships that they had not detected in a first step, they only found 40.07% of the correct relationships; and 32.28% of the relationships they proposed were incorrect.

This section describes the framework that has been used to identify and analyse temporal expressions inside texts.

6.3.1 A Framework for identifying time expressions

Figure 6.3 shows the general architecture of the module aimed at finding events in texts, together with temporal expressions. Times have to be placed in the right places in the timeline, and they have to be attached to the events they modify. During initialisation, the system creates a timeline where any absolute time point or time interval can be placed. The steps are:

1. Preprocessing of the texts: chunk parsing, word-sense disambiguation and identification of narrative events.

2. Identification of events (section 6.3.2).

3. Identification of temporal expressions in the text, and resolution in the timeline (section 6.3.3).

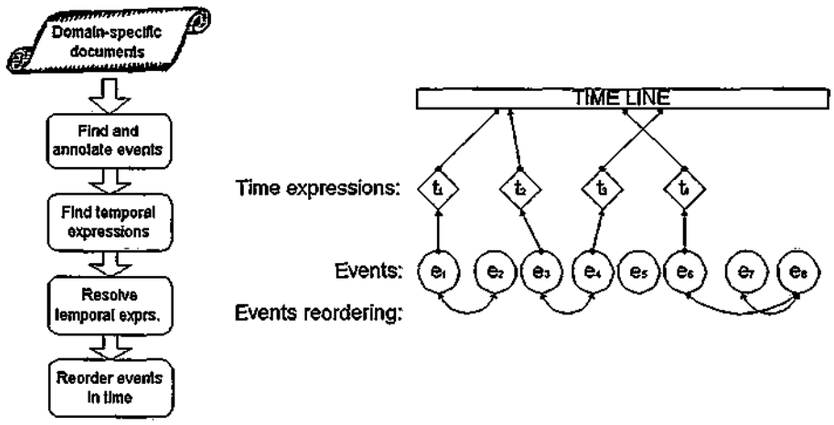


Figure 6.3: General architecture of the system.

6.3.2 Detection of events in a text

At present, most systems that attempt to resolve temporal information from texts do not extract events [Mani and Wilson, 2000] [Koen and Bender, 2000]; or they only consider events denoted by verbs heading syntactic clauses [Filatova and Hovy, 2001]. In the approach described here, the definition of event is extended to include:

- Complements of predicative verbs such as *to be* or *to become*, which usually represent states, e.g. sentence (11a).
- Verbs with lexical information, i.e. excluding auxiliary, modal and predicative verbs; these verbs represent actions or states, e.g. sentence (11b).
- Nouns that, in the WordNet taxonomy, are hyponyms (specifications) of one of the three concepts *act*, e.g. *arrival* in sentence (11c); *event*, e.g. *accident* in (11c); or *state*, e.g. *danger* in (11d).

(11) a. He was **strong** at the time.

b. The train **arrived** on time.

c. The **accident** was **known** before his **arrival**.

d. The general was in **danger**.

Because some nouns may have some senses that represent events and some that do not, it is necessary to use a word sense disambiguation procedure that decides with which sense a noun is used in a given context. These experiments were done with the baseline WSD procedure that assigns, to each word, the sense with which it is more frequent in the SEMCOR sense-tagged corpus [Landes et al., 1998], but it must be noted that accuracy could be increased if a better WSD algorithm was used.

Results detecting events

In order to test these criteria on different kinds of texts, four small texts from different domains have been used: one newswire article from the MUC-3 training data, about terrorism; the first sentences from the Wall

Corpus	Events (human)	Events (system)	Correct	Recall	Precision
MUC-3	66	65	59	89.39%	90.77%
LOTR	67	66	59	88.06%	89.39%
WSJ	73	77	67	91.78%	87.01%
IBM	78	73	68	87.17%	93.15%
Total	284	281	253	**89.08%**	**90.03%**

Table 6.3: Results (precision and recall) for the identification of events in the sample texts.

Street Journal (WSJ) corpus; sentences from a corpus with IBM program instructions, all of them obtained from the Penn Treebank version II [Marcus et al., 1993]; and the first sentences from the first chapter of *The Lord of the Rings* (LOTR) [Tolkien, 1968]. These texts were processed using the linguistic processing tools enumerated in Section 5.3.4. The resulting events were compared to the annotation produced by a human.

The metrics used for measuring its accuracy are **recall**, the fraction of correct events that were found in the test corpus; and **precision**, the fraction of the events proposed by the system that were correct.

The overall results detecting events in the text are shown in Table 6.3. Although these results are much better than the ones obtained by Filatova and Hovy [2001] (a recall of 60.76% and a precision of 55.81%) it must be noted that the corpora are different, and the annotation guidelines are probably dissimilar as well, so our respective results are not really comparable.

The program attains similar results in the first three documents, most of the errors being due to part-of-speech tagging and parser mistakes. The recall of the IBM manual is lower because most of the verbs in the section titles appeared capitalised, and the part-of-speech tagger mistagged them all as proper nouns.

6.3.3 Anchoring events in time

When a writer narrates a story, the events that he or she is telling can be classified in three groups: past events, present events and future events. Every event in the first group will be told in past tense, every current event will be told in present tense, and every future event will be told in future tense. Occasionally, some verbal times can be used in other forms, such as those in sentences (12a) and (12b). Usually, when this phenomenon happens, there are explicit time expressions (e.g. *in 1492* and *tomorrow*) that show that the tense of the verb is not being used in the usual way.

(12) a. After many days navigating, Colon finally **discovers** America in 1492.

b. Tomorrow I **go** to see my parents.

This can be further complicated by the fact that we can embed narrations inside narrations, for instance, when a person in a novel speaks. In that case, the verb tenses in the person's utterances are relative to the time at which that person speaks. We can call **anchor_time** the time at which the narration happens, and **anchor_rules** the criteria used for finding temporal relationships from a verb and the anchor_time using its tense. If, while processing a text, we find an event of *speaking* that introduces a new narrative context, we use the time of that event as the new anchor_time for that context.

The **anchor_rules** are necessary because, when the embedded narration is not quoted literally, but stated as a subordinate clause introduced by the complementiser *that*, then present tenses and future tenses are transformed into past and conditional tenses, so we can recognise which verb tenses refer to the time when

Sentence	*go* happens after...
(13a)	the event of saying
(13b)	yesterday
(13c)	the event of saying
(13d)	now
(13e)	the event of saying

Table 6.4: Temporal relation between *going* and other events in the example sentences in (13).

Process(document)
1. Assign the document's date to a point in the timeline (the anchor_time).
2. anchor_rules ← relations between each verb tense and the anchor_time.
2. Find_tenses(anchor_time, anchor_rules, document)

Find_tenses(anchor_time, anchor_rules, document)
1. Find temporal expressions in the text.
2. Use these temporal expressions to place a few events in time.
    2.1. If they are absolute expressions, place the events in the timeline.
    2.2. If they are relative to the anchor_time, place them accordingly.
3. Use the anchor_rules to find relations between the verbal events and the anchor_time.
4. Find temporal relations between events, using features such as prepositions (*before*, *after*, etc.).
5. For every *narration event*,
    5.1. fragment ← portion of the document with the narration.
    5.2. new_anchor_time ← the time when the narration event takes place.
    5.3. Create the new_anchor_rules for the fragment:
    5.4. If the narration is a literal citation, tenses are relative to new_anchor_time.
    5.5. Else,
        5.5.1. Past and conditional tenses are relative to new_anchor_time, as present and future.
        5.5.2. Present and future tenses refer to the current anchor_time.
    5.6. Find_tenses(new_anchor_time, new_anchor_rules, fragment)

Figure 6.4: Pseudo-code of the algorithm for finding temporal relations between events.

the utterance was produced and which refer to now. Table 6.4 shows when the event of *going* is interpreted to happen for each of the sentences in (13).

(13) a. Peter told me yesterday, 'John said that he would go'.

b. Peter told me yesterday, 'John said that he will go'.

c. Peter told me yesterday that John said 'I will go'.

d. Peter told me yesterday that John said that he will go.

e. Peter told me yesterday that John said that he would go.

The algorithm is displayed in Figure 6.4. Initially, the document has to be placed in the timeline, at the initial anchor point. If the document contains a header specifying its date, then we can place it at a fixed point in the timeline; otherwise, we just assign to it an unknown time point. Every event that is described in past-tense will be placed before the document's time. And, inversely, future-tense verbs will describe events that happen after the writing of the document. Expressions such as *today* or *three months ago* also help in placing them in the timeline.

Next, time expressions are identified in the document and the events they modify are placed accordingly in the timeline. After that, prepositions and conjunctions introducing event clauses are used to find the

Corpus	Total (human)	Found (system)	Correct	Recall	Precision
MUC-3	11	9	8	72.72%	88.89%
LOTR	3	3	3	100.00%	100.00%
WSJ	11	10	10	90.91%	100.00%
IBM	0	0	0	-%	-%
Total	24	22	21	**87.50%**	**95.45%**

Table 6.5: Results for finding temporal expressions in the documents

ordering of the remaining events. Finally, if there is any event of narrating, it is processed, but the anchor point for the events inside the narration will be the time when the narration occurred.

Identifying time expressions

Writers place events in time using **Time-denoting expressions** [Móia, 2001], such as `three days ago`. These expressions refer to intervals in time. The system uses a hand-coded list of regular expressions in a FLEX file to recognise absolute time expressions (e.g. dates) and relative expressions (e.g. *today*). These expressions can be easily used to place the events modified by them in the timeline. In the experiments, a total of 22 expressions were found, giving a recall of 87.5%, and a precision of 95.45%, although the set is too small for these results to be significative. Table 6.5 shows the results obtained by the regular expression recogniser when applied to our small documents. Although the test corpus is small, it can be appreciated that, because of the way it was built, the regular expressions have a high precision, and only one result was judged incorrect because it only returned a part of a more complex expression. Recall is a bit lower, because the coverage of the regular expressions can be improved.

After *time-denoting expressions* have been found, we use prepositions and conjunctions to translate them into absolute or relative time-intervals. Some time expressions, such as *today* or *tomorrow*, or time prepositions such as *ago* are relative to the anchor time; while other time expressions such as *before* or *after* are relative to the time of the event in the main clause. All this is taken into account.

In the sample texts, the simple heuristic of assigning the time to the nearest event in the same syntactic clause was 100% correct for the 21 expressions found in the text. It must be born in mind that the number of time expressions is very small and not significative, so we still cannot draw conclusions.

Finally, Table 6.6 displays the number of temporal relationships that were discovered using the anchor rules. The first column shows the total number of relationships discovered, and the second column shows the number of inferences from the previous, by applying the symmetry of the *simultaneous* relationship and the transitivity of the *preceding* relationship:

$$t_1 \; \textit{Simultaneous} \; t_2 \Leftrightarrow t_2 \; \textit{Simultaneous} \; t_1$$
$$(t_1 < t_2) \wedge (t_2 < t_3) \Rightarrow (t_1 < t_3)$$

In the LOTR corpus there were few inferences, because most of the verbs were in past tense (preceding the anchor time) and there were few temporal expressions with which place the events in the timeline.

6.4 Identification of domain-dependent terminology

In different kinds of texts we may find that it is useful to recognise different kinds of terminology, such as personal names, names of businesses and governmental organisations, chemical compounds, etc. In some

	Tense relations	
Corpus	**Found**	**Expanded**
MUC-3	61	911
LOTR	51	183
WSJ	95	1226
IBM	32	1083

Table 6.6: Results for finding temporal relations between events, using the verb tenses and the temporal expressions.

cases, it is possible to build high-accuracy specialist modules for finding them, so it is also possible to identify and classify low-frequency terms.

The architecture of the system, where the text processing modules are organised as a pipeline, allows as many of these modules as necessary, and it has been tested with a module that automatically finds scientific or Latin names of life forms. Its recognition does not pose big difficulties but, to the author's knowledge, it is a topic that has not received much attention, so it will be discussed in this section.

Scientific names always follow the same pattern: the name of the genus, capitalised; the name of the species, in lowercase; and, optionally, the name of the subspecies and an abbreviation of the name of the person that named the life form. Quite often, only the initial of the genus name is written, if it can be inferred from the context. When this happens, it is always written uppercase. On the other hand, when the name of a species is not relevant or unknown, it is usually written as the abbreviation of *species*, sp.

Other fact that can be taken into consideration is that scientific names usually have Latin or Greek roots; although it is not always the case. In some occasions, they contain the name of a biologist, such as *Lystrophis d'orbignyi* for Alcide d'Orbigny or *Polybius henslowi* for John Stevens Henslow, which contain a French and an English proper name, respectively; or the name of the place where it was discovered. But even in these cases, the non-Latin words are inflected so they have Latin suffixes.

From this fact, we can draw two conclusions:

- The endings of the genus and species are nearly always typical Latin suffixes. Furthermore, the genus name is usually in nominative case, while the species appears frequently both in nominative and in genitive. Therefore, it is possible to compile a list of the most frequent endings of scientific names, in order to identify these in a text. Table 6.7 shows the typical endings both for the genus and species of scientific names, which was compiled from a list with around 300 items.
- Scientific names are usually written in Latin. Therefore, using a language identification module should be a help for identifying scientific names in a text.

6.4.1 A procedure to identify automatically scientific names

Taking all the previous information into account, the following procedure was used to recognise scientific names:

1. Extract all pairs of words such that the first one is capitalised and the second one is lowercase.
2. Filter out all the pairs that contain a common English word, inflected or not.
3. Check that the endings of the words are valid endings.

Gender endings					
a	e	i	oe	u	yps
an	eps	il	on	um	ys
anx	er	is	ops	ur	yx
as	es	ix	or	us	
ax	ex	o	os	ynx	

Species endings					
a	ax	es	o	ox	ys
ae	e	ex	on	u	
ans	ens	i	ops	um	
ang	eps	is	or	ur	
as	er	ix	os	us	

Table 6.7: Typical endings for the genus and the species of a scientific name.

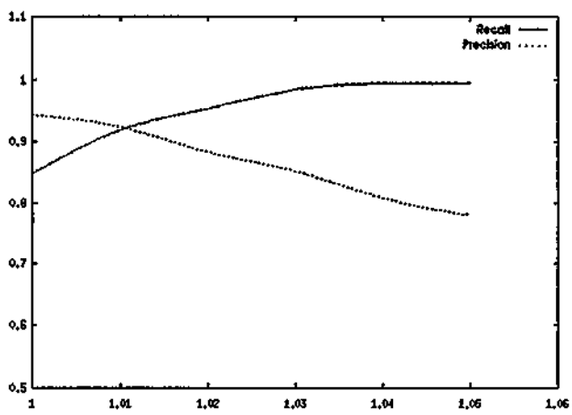


Figure 6.5: Recall-precision curve for identifying the language of the first 200 pairs of words from *The Voyage of the Beagle* and 200 scientific names.

4. Use a language-identification module to make sure that the words are Latin.

5. Finally, for every genus that has been identified, mark it as well whenever it appears alone in the texts (without the name of the species).

The first step selects all pairs of words such that the first one is capitalised and the second is not. Therefore, not only scientific names but also most of the sentence-initial pairs of words were selected as well (e.g. *I was*); together with all the proper nouns that were followed by a common word (e.g. *Tierra del*, from *Tierra del Fuego*). The second step filters out most of the sentence-initial pairs of words. The function of steps 3 and 4 is to retain only the scientific names. Finally, step 5 marks those scientific names that only include a known genus name.

6.4.2 Results

The different filters have been evaluated separately, to study their impact on the overall precision and recall. The language identified was evaluated on 200 random scientific names, and the first 200 pairs of words

Threshold	Recall	Precision
1	.850	.9444
1.01	.920	.9246
1.02	.955	.8843
1.03	.985	.8528
1.04	.995	.8089
1.05	.995	.7804

Table 6.8: Recall and precision values for classifying pairs of words as Latin or English. The pairs of words were 200 scientific names, and the first 200 words from *The Voyages of the Beagle*.

from *The Voyages of the Beagle*, which did not contain any scientific name. Only the language identification module is used in order to determine whether the word pairs were written in English or in Latin. The module used was `text_cat`, written by Gertjan van Noord according to the N-gram-based algorithm by Cavnar and Trenkle [1994]. This algorithm calculates frequencies of N-grams from the text to be classified, and these frequencies are compared to the profiles for different languages. In this case, only English and Latin are considered.

For classification, the program calculates the ratio between the distance from the profile of the text to be classified to each of the profiles of the two languages. If this ratio is below a given threshold, both languages can be equally plausible; while if it is over the threshold, the language with the smallest distance is returned as the answer. For example, let us suppose that the distance between the profile of the text and the English profile is 100, and the distance between the profile of the text and the Latin profile is 104.5. The ratio between both values is $\frac{104.5}{100} = 1.045$. If the threshold is set to any value below 1.045, then the text can be classified as English; while if the threshold is over that value, the program will answer that it could be any of the two languages.

This experiment allowed to find which was the best threshold to use. For the experiment, it was considered that two words could be classified as scientific names if the program returned either *Latin* or *any of the two*. Therefore, if the threshold is large, then there are many possibilities that the words will be classified either as Latin or as *any of the two*. In this case, the recall will be high, but the precision will be low. On the other hand, if the threshold is small, then there are less possibilities that the words will be classified as Latin, and therefore precision will be high but recall will be low.

The above-mentioned 400 pairs of words, of which 200 were scientific names, were classified. Figure 6.5 shows the recall-precision curve for different values of threshold from 1 to 1.05, and Table 6.8 shows the actual values obtained for precision and recall. The threshold that was taken was 1.02, which produced a recall over 95%, and still with an acceptable precision.

Secondly, we can evaluate the impact of the other two filters in this procedure (that the words cannot be present in English dictionaries, and that the endings of the words have to belong to the lists). The language identification was combined with the two filters, and, when applied to the list of 400 pairs of words, the results were the following:

- 196 scientific names out of the 200 were correctly identified, which represents a recall of 98%.
- For the remaining 4 scientific names, the genus was correctly identified and classified as a genus; but the species was not accepted because it was equal to a common English word: *simile*, *glycine*, *major* and *martini*.

- After filtering all the word pairs, all the scientific names proposed were correct, and therefore precision was 100%. It must be noted, however, that the 200 pairs of words that were not scientific names consisted entirely of English names; this algorithm might fail if, for example, it were applied to an English text containing a portion written in Spanish or in Latin, because it might classify that portion of texts as scientific names incorrectly.

6.5 Summary and discussion

The output of the off-line processing of the system consists of the original, linear texts annotated with linguistic information, and a database with additional information about the text, such as the semantics of the unknown terms, and about dates and other terminology found in them.

6.5.1 Term Classifications

The algorithm for Term Classification in a lexical ontology described in the previous chapter can be improved by finding *hyponymy patterns* in domain-specific texts. These patterns consist of regular expressions of words, or syntactic dependencies between them that usually express a hyperonymy relationship. This chapter described a framework for integrating the two different algorithms, the one based on the contextual analysis and the Distributional Semantics techniques, and the one based on hyponymy patterns. The obtained result is a more robust classification algorithm, as all evaluation metrics have improved.

The top-down classifier, based on the context words, suggests a path from the root of the ontology down to the concept that will be suggested as the maximally specific generalisation of an unknown concept. The hyponymy patterns help this algorithm in selecting a concept when the context does not give much information, such as for male-female distinctions.

The result is a deterministic unsupervised system that also allows the attachment of new concepts to any intermediate level in an ontology, not only at the leaves. It has been shown that it is able to tackle big ontologies with the size of WordNet.

Because it does not require any previous hand-coding of patterns, and the concept contexts are also automatically collected from the Internet, it could possibly be ported to other languages, if the syntactic processing tools used are available for them.

Dates identification

It is also possible to identify events and time expressions in the texts, and to reconstruct, to some extent, the chronological ordering of the events from a text. The algorithm described here seems to be general enough to be used with documents from different domains, such as political news, financial news, novels and user manuals.

The method built for identifying events has two advantages over previous approaches: WordNet is used to identify common nouns that can represent events (Section 6.3.2), and it allows different events to be in the same clause, even if they happened at different times. Therefore, it is not necessary to have a very accurate parser for the experiments.

Next, a new algorithm was described for resolving the time expressions, regardless of whether they are absolute calendar dates, relative to the time the text was written, or relative to the event in the main clause. The advantage of this algorithm is that it deals easily with embedded narrations, by updating a set of

anchor rules that keep track of the semantic meaning of verb tenses in each case. This is specially useful for processing newswire articles and novels, because both include speeches quite often.

It must also be noted that there are some time expressions which still have to be disambiguated, such as the two meanings of the word *today*. This word can carry the general meaning, similar to *nowadays*, or the specific meaning *the current day*. In this work, only the specific meaning is considered; although some approaches have been proposed to distinguish them. Mani and Wilson [2000] used a decision tree classifier for distinguishing different senses of time expressions.

The extension of the events with information about the agents and objects of the action events can also have other applications, such as for Question Answering systems. For example, if a user asks questions such as (14a) and (14b), it is necessary to do some kind of temporal processing of the texts to be able to answer.

(14) a. *How many times did Frodo visit Lorien?*

b. *How often has the current president of Portugal travelled abroad?*

Identification of specific terminology

The architecture allows us to add optional modules for identifying specific terminology. Currently, it has been tested with a module that looks for scientific names of life forms. Because of the way these terms are constructed, a language identification module seemed a very good solution for this problem, combined with some filters that contrast Latin endings and ignore English words, in order to keep just the scientific names.

The results show that, if an English text only contains English and scientific names, these can be recognised with high accuracy. It is not clear whether this would work in texts containing other languages, as romance languages are more hardly distinguished from Latin with automatic procedures. Particularly, if an English text is processed that includes portions written in Latin or in other Romance language, it would be necessary to adapt the algorithm in order to ignore these portions.

Part III

On-line processing: Text generation

Introduction to Part III

During the off-line processing for building the adaptive hypermedia site, one or several source texts have been analysed, the relevant terms have been found, and the set of texts has been annotated with information about the sections, syntactic analysis, temporal information, etc. A database has been generated with all the information required so it is already possible to connect to the web interface and retrieve the information from the texts, in a way that is adapted to the interests of the user.

This adaptation takes into consideration several features of the users. One of them is their interest in the different topics that are discussed in the text: in this case, they can select pre-defined topics, or create their own personal profiles. A text summarisation algorithm has also been used so they can restrict the amount of information that they will read.

This part starts with a summary of the current work on text summarisation, with special interest on the procedures that are more similar to the summariser that has been built for the adaptive site.

Next, it describes the characteristics of the architecture that has been designed, and implemented in the WELKIN system. It starts with a description of the users profiles, which are used to decide which information will be presented to them. The part finishes with a description of the user interface, and methods to extend the original information with additional data that is automatically downloaded from the Internet.

Chapter 7

Automatic Text Summarisation

Once the original documents have been annotated with linguistic and temporal information, and the relevant terms have been extracted and classified into the lexical ontology, the system is ready to accept users and provide them with information according to their preferences. But this information is not presented to them exactly as it was in the original documents; it is processed so that visitors browse just the information that is more relevant to them. Factors such as users interests may determine the contents of the information shown, and time constraints may require the system to condense the information so it takes less time to read.

Automatic Text Summarisation is the task that consists in finding, in a textual source, which information is more relevant for a user or an application, and presenting it in a condensed way. Summarisation systems can vary largely depending on their final application. For instance, a system that sends summaries of e-mails to mobile phones will be very different from a system that generates summaries of different newswire articles about the same topic, or a system that produces abstracts of scientific papers. Each of these will have to process different language styles and different domains, and the requirements about the readability and size of the generated summaries will not be the same.

The aim of this chapter is to provide an overview of the different algorithms that have been explored up to date. Probably the most comprehensive review on Automatic Text Summarisation systems is the one provided by Mani [2001], from which most of the contents of this chapter have been obtained.

7.1 Introduction

Text summarisation systems can be classified in two different kinds:

- **Text extraction** systems, which cite, literally, fragments of the original text. These fragments may be whole paragraphs, sentences, clauses, words, etc.; it may consist in removing the closed class words (prepositions, determiners, conjunctions, etc.), or in extracting the sentences that are judged more relevant. As the compression level increases, more information is discarded from the original documents.
- **Text abstraction**, on the other hand, consists in summarising the original material, including at least some information that was not present in the original document. Text abstraction usually needs some kind of semantic processing of a text, while extraction can be attained without the use of a semantic processor. For example, the abstract of a newswire article that describes several terrorist attacks may include the number of casualties altogether, even though that certain number may not be present in

the original article; a text extraction system, which merely copies fragments of the source will never provide that information.

In the classification performed by Mani [2001, pg. 13], summarisation systems are classified taking into consideration several parameters. Apart from the relation between the summary and the source (extract or abstract), which has already been discussed, the following ones are relevant for our purposes:

1. **Compression rate**, or ratio between the sizes of the summary and the document.
2. **Audience**, which can be generic or user-focused.
3. **Function**: informative (a synopsis), indicative (highlighting the differences between the sources), or critical.
4. **Coherence**: whether the produced summary must be coherent, or whether it can be allowed to be incoherent. For example, in the second case it is possible to produce a summary by removing prepositions and determiners, because it is still possible for a human to understand the original information; while some applications require that the summary be a coherent text.
5. **Span**: summaries of a single document, or merging information from multiple documents.
6. **Language**: mono-lingual, multi-lingual or cross-lingual (when the summary is written in a language different than the source document). If the summariser only processes texts from a particular domain, then it may be specially adapted to a *sublanguage*, i.e., the restricted vocabulary and syntactic constructions used in that domain.
7. **Genre**: it may be necessary to use different strategies for different varieties (e.g. narrative, newswire articles, technical documents, poetry, etc.).

There is another characteristic that may be taken into account when designing a summarisation system, but which has not been considered relevant for this work, as it is only capable of processing text:

8. **Media**: summarisers can process different media, such as text, tables, images, audio or video. If either the input or the output includes several media types, it is called *multimedia summarisation*.

Concerning the amount of linguistic processing that is performed to the texts, a document summarisation system can be classified as:

- **Morphological**: when the only linguistic processing that is performed is morphological analysis (identification of stem and affixes). For example, a system that selects relevant sentences by looking for keywords (inflected perhaps) that are known to be important for a topic would belong to this category
- **Syntactic**: when the system performs a full parsing of the source documents.
- **Semantic**: when the sentences are translated into a semantic representation language and then they are re-stated using a Natural Language generation module. For example, an Information Extraction system that processes documents about a particular domain and fills in templates with the extracted data, can next use a language generation module to present that data.
- **Pragmatic**: when the system includes a discourse analysis module, e.g. for identifying the antecedents of pronouns; or for making inferences using "common sense" knowledge.

7.2 Text extraction

Text extraction consists in locating the relevant units of a text and extracting them together to construct the automatic summary. These units can be, for instance, words, phrases, sentences, paragraphs, whole sections, etc. For example, for some applications a list with all the proper nouns, or a list with all the nouns and verbs that appear in the main clauses of a text could be considered relevant summaries.

Once the type of the unit has been decided, a popular process followed to produce an extract consists of the following steps: firstly, the unit boundaries are identified (word boundaries can be identified using a tokeniser; sentence boundaries, with a sentence splitter; or phrase boundaries with a shallow parser, etc.). Next, the separate units each receive a relevancy score. Finally, they are ordered according to the score, and the best ones are selected. If the length of the target summary is set at N% of the original document, then the N% units with the highest score are selected, and they are all put together to form the summary.

Note that if the unit is very large, then the summary may be too coarse-grained to be useful. For example, if only whole paragraphs are extracted, then a 10% summary of a text that contains 10 paragraphs can only have one paragraph, and all the remaining information is discarded. On the other hand, if the unit taken is a sentence, then it is possible to extract sentences from different paragraphs and it is possible to produce more varied summaries of the original text.

The procedures for text extraction may perform different kinds of analyses to the texts, from a morphological approach based on term frequency or keywords to detailed discourse analyses. However, regardless of the sophistication of the linguistic analysis performed for extracting the units, these approaches are prone to suffer from the following problems:

1. They extract single discourse units (e.g. sentences) instead of sequences of them. This may lead to incoherence, for instance, when the extracted sentences contain conjunctions at the beginning of sentences, dangling anaphoras, etc.

2. Sentences should not be evaluated independently beforehand, and next extracted; quite on the contrary, the choice of a sentence should, for example, block the choice of a different sentence that describes the same idea. Otherwise, it may be the case that the top two sentences are each a paraphrase of the other.

In order to improve the readability of the extract, and to mitigate these problems, extracts are usually post-processed. The following are some typical problems that can be worked out at a post-processing phase:

1. Lack of conjunctions between sentences, or dangling conjunctions at the beginning of the sentences (e.g. an extracted sentence starting with *However*). These conjunctions can be eliminated after the sentences have been extracted.

2. Lack of adverbial particles (e.g. it would be desirable the addition of *too* to the extract *"John likes Mary. Peter likes Mary"*).

3. Syntactically complex sentences.

4. Redundant repetition (e.g. repetition of proper names, which can be substituted by pronouns).

5. Lack of information (e.g. complex dangling anaphoras, such as *in such a situation*; or incomplete phrases).

The following sections contain a review of some of the approaches that are used for text extraction.

7.2.1 Edmundsonian paradigm

Many extraction systems are based on the work by Edmundson [1969], which described a procedure that has later received the name of *Edmundsonian paradigm* [Mani, 2001]. This paradigm defined the way in which each unit is given a score as a linear function of several features.

Working with papers about chemistry, Edmundson collected four sets of words: *title words*, or words taken from the title of the articles and the headings; *bonus words*, the set of words that appear in the collection of chemistry documents with a frequency that is over a certain threshold; *stigma words*, or the set of words that appeared below other threshold; and, finally, for each separate document in the collection, a separate list of *key words* is collected, consisting of a document-specific set containing all the words that appeared above a threshold frequency in the document, but which were not *bonus* or *stigma words*.

These sets were used to weight the different sentences of the documents. If we call $C(s)$ the weight of a sentence due to the *cue words* it contains (bonus and stigma words), $K(s)$ the weight due to the key words; $T(s)$ the weight due to the title words that it contains; and $L(s)$ a weight that depends on the position of the sentence in the text –sentences that appeared right below section headings and the first and last sentences in every paragraph received a high mark–; then the final weight was calculated as a linear function of the separate weights:

$$W(s) = \alpha C(s) + \beta K(s) + \gamma L(s) + \delta T(s) \tag{7.1}$$

This procedure can be extended with more weight functions, or with a reinterpretation of these functions. So, for example, $C(s)$ can also consider the presence or absence of emphasis words and phrases (words that usually introduce important statements, such as *in conclusion*); and the keyword weight function $K(s)$ can be constructed using the tf-idf metric (see Section 4.2.1). Other works (e.g. [Lin and Hovy, 1997, Mani and Bloedorn, 1998]) further extended this set of features.

Instead of tuning the parameters, α, β, γ and δ, by hand, it is also possible to learn a model using Machine Learning procedures, in order to weight the document sentences. Under this framework, it is necessary to manually create an extract for each document in the training data. Then the values of the functions, C, K, L and T would be calculated for each sentence to create a vector, and then each vector would be labelled as a positive example if the sentence had been included in the extract, or as a negative example otherwise. These examples can be used to learn a model to weight sentences, which can be applied to a test set for evaluation purposes.

The Edmundsonian paradigm has often been criticised for the following reasons [Mani, 2001]:

- It does not take into account the compression rate for the extraction process. For example, if the top two sentences, s_1 and s_2, provide together an idea; and the third sentence in the ranking, s_3, provides alone other important idea, then a 1-sentence summary should select s_3, because it includes a complete idea, than either s_1 or s_2 alone.
- The linear model might not be powerful enough for summarisation. Furthermore, the equation does not tell us, theoretically, why the sentences are chosen.
- Finally, it only uses morphological information; whereas syntactic and semantic analysis are obviously important in order to produce a high quality summary.

However, for some applications, the Edmundsonian model is powerful enough, and it can be extended in order to include some syntactic information and implemented with a post-processing for improving the coherence

of the extracts.

7.2.2 Summarisation using discourse analysis

We have seen that one of the problems of the Edmundsonian paradigm is that the resulting extracts may be incoherent, and a post-processing approach might not be sufficient or might come too late to repair the extract. Approaches that try to capture discourse structure in the original texts try to prevent this problem. Mani [2001] distinguishes two ways in which discourse can be studied:

- **Text cohesion** [Halliday and Hasan, 1996] involves relations between words, word senses or referring expressions, which determine how tightly connected the text is.
- **Text coherence** [Halliday, 1978, Mann and Thompson, 1988] represents the overall structure of a multi-sentence text in terms of macro-level relations between clauses or sentences.

Text cohesion studies how related the words in a document are. These relationships or *ties* can be of any kind: morphological (e.g. reiteration, or words that share the same stem); co-occurrence relationships (e.g. words that appear consecutive); syntactic (e.g. coordination); semantic (e.g. synonymy, hyperonymy); or discourse relations (e.g. anaphora, ellipsis). Thus, a text is very cohesive if there are many such ties between its words.

Experimental studies [Irwin, 1980] show that people tend to remember better highly cohesive texts; furthermore, when a text contains 'macro-statements', which synthesise information from more than one statement, these sentences are easier to remember. This suggests that high-cohesion passages may be easier to summarise.

Summarisation systems that are based on cohesion take into consideration the ties that join the words from different units (paragraphs, sentences, or clauses). We can say that a unit c_1 is related to other unit c_2 if a word inside c_1 has a tie to a word inside c_2. For instance, if c_1 and c_2 contain two words which are synonyms, then there is a tie between them. If we represent each sentence as a graph node, and we consider these relationships as non-directed arcs, we can construct a graph representation of the whole text.

Skorochod'ko [1972] stated the following criteria in order to calculate the salience of a sentence in a text:

> **Connectivity criterion:** The salience of a sentence is proportional to the number of sentences that are semantically related to it.
>
> **Indispensability criterion:** The salience of a sentence is proportional to the degree of change to the graph when the sentence is removed.

He specified the following formula to combine these two factors:

$$F_i = N_i(M - M_i) \tag{7.2}$$

where F_i is the salience of sentence i; N_i is the number of edges incoming to sentence i; M is the number of nodes in the graph, and M_i is the maximum number of nodes in any connected components if i is removed from the graph. For example, in the graph from Figure 7.1, sentences 4 and 5 have a salience of 16 and 25 respectively.

Similar approaches are those described by Salton et al. [1997] and Mani and Bloedorn [1999].

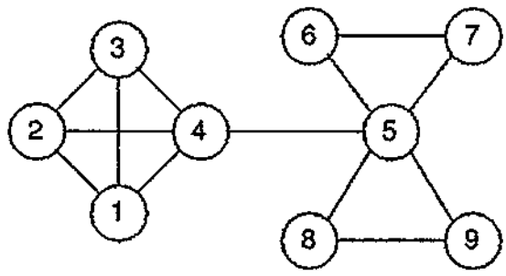


Figure 7.1: Example cohesion graphs.

On the other hand, **text coherence** is a discourse-level theory that describes the structure of a text in terms of relations between clauses or sentences. Using the following example from Hobbs [1985],

(15) a. John can open Bill's safe

b. He knows the combination,

he claims that sentence (15a) is related to sentence (15b) via a relationship that he calls 'elaboration'; otherwise, the sequence of both sentences would be incoherent.

There are many different theories for coherence, including Rhetorical Structure Theory [RST, Mann and Thompson, 1988], Discourse Grammar [Longacre, 1979], Coherence Relations [Hobbs, 1985], etc. This review shall focus on RST, one of the theories for which there are computational theories and implementations.

RST defines a set of about 25 discourse relationships, such as the ones displayed in Table 7.1. A characteristic of this theory is the fact that many of the binary relationships between clauses are considered asymmetric, i.e., one of the clauses is ancillary to the other. In these cases, the central idea is called the *nucleus*, and the other, which tends to enhance the function of the nucleus, is called *satellite*. There are also a few symmetric relationships, such as the **joint** relation that joins several co-ordinated clauses.

Marcu [1999] describes an approach to parse a set of clauses with RST relationships. He observes that some cue phrases are used to overtly express the relation that holds between clauses. For example, a **purpose** relation is often expressed by means of the phrases *so as to* or *in order to*; and a **concession** relation by means of the conjunction *although*. Using these cue phrases and a Machine-Learning approach, Marcu built a rhetorical parser that constructs the rhetorical structure of a multi-clause text. As an example, Figure 7.2 shows one of the texts that was analysed, and Figure 7.3 shows the parse tree with the relationships between the clauses.

The application for Text Summarisation consists in the following fact: the root node in the rhetorical parse tree is supposed to contain the most important proposition in the whole text. So, in order to generate a very short summary, that proposition would be the one extracted. For instance, the root node in the tree in Figure 7.3 indicates that the whole text can be represented with clause number 2, i.e.

Mars experiences frigid weather conditions.

If we wanted a summary somewhat longer than that, we would just go down one or several levels in the parse tree, and select the nucleus of the different nodes. In the example above, if we go down one level in the parse tree, we could produce a two-sentence summary:

Relation name	Definition	Example
Circumstance	The satellite sets a temporal, spacial, or situational framework in the subject matter within which the reader is intended to interpret the situation presented in the nuclear text span.	As your floppy drive writes or reads, *a Syncom diskette is working four ways to keep loose particles and dust from causing soft errors, dropouts.*
Motivation	The nucleus is an action performable but not yet performed by the reader. The satellite describes the action, the situation in which the action takes place, or the result of the action in ways that can help the reader associate value assessments with the action. The value assessments must be positive, to lead the reader to want to perform the action.	*Now, buy a specially marked box of 10 Memorex 5 1/4" mini flexible discs and we'll send you an additional mini-disc FREE.* Features like our uniquely sealed jacket and protective hub ring makes our discs last longer. And a soft inner liner cleans the ultra-smooth disc surface while in use. It all adds up to better performance and reliability.
Purpose	The satellite presents the effect intended by the actor of the action presented in the nucleus.	*We repeatedly are told we have to move* to hit the ball — but it's just as important to move after you hit it.
Solutionhood	The nucleus is presented as a solution to the problem posed in the satellite.	What if you're having to clean floppy drive heads too often? *Ask for Syncom diskettes, with burnished Ectype coating and dust-absorbing jacket liners*

Table 7.1: Four rhetorical relations from Mann and Thompson [1988], taken from Mani [2001]. The text in *italics* contains the nucleus of the relation.

> Mars experiences frigid weather conditions. Most Martian weather involves blowing dust or carbon dioxide.

In general, depending on the compression rate needed, we could proceed downward in the parse tree, and select more sentences.

7.3 Text Abstraction

An *abstract* differs from an *extract* in that it contains some information that was not present in the original text. An abstract contains inferences made from the text, possibly referring to "common-sense knowledge" or "background knowledge". Mani [2001] structures the process of abstracting a document in three steps:

- Building a semantic representation for the sentences.
- Performing selection, aggregation and generalisation on the semantic representation to condense it while keeping the most relevant information.
- Generating natural language from the condensed semantic representation.

[With its distant orbit { — 50 percent farther from the sun than Earth — } and slim atmospheric blanket,[1]] [Mars experiences frigid weather conditions.[2]] [Surface temperatures typically average about -60 degrees Celsius (-70 degrees Fahrenheit) at the equator and can dip up to -123 degrees C near the poles. [3]] [Only the midday sun at tropical latitudes is warm enough to thaw ice on occasion,[4]] [but any liquid water formed that way would evaporate almost instantly [5]] [because of the low atmospheric pressure. [6]] [Although the atmosphere holds a small amount of water, and water-ice clouds sometimes develop, [7]] [most Martian weather involves blowing dust or carbon dioxide. [8]] [Each winter, for example, a blizzard of frozen carbondioxide rages over one pole, and a few meters of this dry-ice snow accumulate as previously frozen carbon dioxide evaporates from the opposite polar cap. [9]] [Yet even on the summer pole, {where the sun remains in the sky all day long,} temperatures never warm enough to melt frozen water. [10]]

Figure 7.2: Clausal analysis of Mars text [Marcu, 1999].

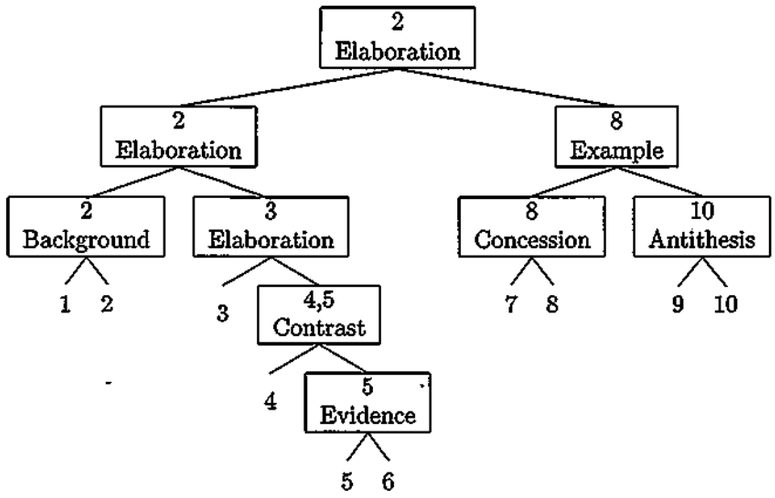


Figure 7.3: RST-analysis of Mars text from Figure 7.2, showing promotion [Marcu, 1997]. The numbers in the rectangles show which children are the nucleus of the relationships.

7.3.1 Abstracting with Information Extraction

A natural way of building an abstract generator under the proposed framework consists in using an **Information Extraction** (IE) system. The purpose of these systems is to fill pre-defined templates (for a given domain) with information found in a document [MUC6, 1995, MUC7, 1998]. For example, a system that analyses newswire articles in the domain of terrorism may be able to find in the documents, for each terrorist attack, the date and location of the event, the names of the terrorists and the terrorist organisations involved; the number of people dead and injured; and the damages on buildings and infrastructures. Because the output of the system is highly structured, it is possible to create a domain-specific IE system with an accuracy good enough for practical applications.

Let us consider the newswire article from Figure 7.4, in which one terrorist event is told. A possible output template would be the one in Figure 7.5, and once the template has been filled in, a text generation component could produce the output. In this approach, the semantic representation of the text would be the template with all its values, which contains only the more relevant information (the one that was considered

LIMA, 16 JAN 90 (TELEVISION PERUANA) – Ten terrorists hurled dynamite sticks at U.S. Embassy Facilities in the Miraflores district, causing serious damage but fortunately no casualties. The attack took place at 21:00 on 15 January [01:00 GTM on 16 Jan].
Inside the facility, which was guarded by 3 security officers, a group of embassy officials were holding a group meeting.
According to the first police reports, the attack was staged by 10 terrorists who used 2 Toyota cars which were later abandoned, one of the vehicles was left on the third block of Jose Pardo avenue, while the other was left on the first block of Bella Vista street in Miraflores.

Figure 7.4: Text for information extraction, from the Third Message Understanding Conference [Sundheim, 1991].

DATE	15 jan 1990
TIME	21:00
LOCATION	Lima
TERRORISTS	"10 terrorists"
TERRORIST ORG.	-
CASUALTIES	0
INJURED	0
PHYSICAL DAMAGE	"serious damage" on "the U.S. Embassy"

Figure 7.5: Template generated for the text in Figure 7.4

important enough as to be included in the templates). The generation module may consist of a pretty print of the template, or a simple text generation module that follows some script to translate the template into a natural language text.

The templates in the MUC conferences were pre-defined, depending on the topic of the text collection, and the interests of the sponsors, such as news agencies. Therefore, IE-based systems are usually domain-dependent and cannot be applied to general texts. However, there have recently been some advances in general-purpose IE. Harabagiu and Maiorano [2002] describe a framework with which it is possible to generate the target template from a collection of documents; if the template is not available, the definitions from WordNet are used to locate topical relationships between the domain-specific terms (terms that appear in the definition of other term are considered to share the same topic); and some of them are selected, based on a frequency analysis, to be part of the template for IE. Harabagiu and Maiorano apply their system both for multi-document summarisation and general-purpose IE.

7.3.2 Abstracting by identifying events

A second approach to abstracting consists in identifying events in the text, and finding the relationships between them, in order to locate which are the most relevant ones, the ones that should appear in the abstract. A procedure to select the most relevant events consists in building a conceptual graph, where nodes represent events and arcs represent relations [Alterman, 1985, Alterman and Bookman, 1992]. Arcs can represent temporal relations (e.g. *precedes*, *simultaneous*), causing and enabling relations, hyperonymy relations (*subclass*), meronymy relations (*part-of*), etc. As an example, the portion of a semantic network

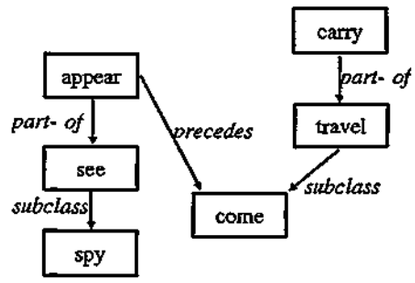


Figure 7.6: Concept graph where the two events from sentence (16), *spy* and *carry*, are related through arcs.

that is relevant for sentence (16) is represented in Figure 7.6.

(16) The prince **spied** the pig **carrying** her laundry to the stream.

Alterman and Bookman consider that only the events such that they are root nodes in the graph –nodes with no incoming arcs– and such that the number of nodes reachable from them is above a given threshold are relevant for the summary. In the graph in the figure, **appear** is more relevant than **carry** because there are more nodes accessible from it.

The relevancy of an event E can also be calculated based on a frequency analysis. Maybury [1995] used the following metrics in order to decide whether an event is relevant enough to appear in the abstract:

- The relative frequency of an event, which can be calculated as

$$RelEventFreq(E) = \frac{count(E)}{\sum_{e \in Events} count(e)}$$

- The relative number of relationships between events in which E is involved. These relations include causing and enabling relations; the intuition is that an event that causes many other events must be important:

$$RelRelFreq(E) = \frac{W(E)}{\sum_{e \in Events} W(e)}$$

$$W(E) = \sum_{r \in relations(E)} significance(r) \cdot count(E, r)$$

Other domain-specific importance measures can also be taken in consideration, such as events that are known to be always important in a particular domain.

7.3.3 Abstracting with internal semantic representations

A third approach to abstraction consists in producing a semantic representation of the sentences in the document (for example, in FOPL) and next performing selection, aggregation or generalisation operations on this representation. A drawback of these systems is that the step of producing a semantic representation of unrestricted texts involves many problems that have still not been fully solved, such as Word Sense Disambiguation (WSD) or coreference resolution, although advances are being made.

One of the approaches that follow this method makes use of macro-rules [van Dijk, 1979, 1988] for producing the meaning representations. The procedure is the following:

Operation	Example	result
Deletion	Peter saw a blue ball	
	i.e. Peter say a ball, the ball was blue	⇒ Peter saw a ball
Generalisation	Peter saw a hawk	⇒ Peter saw a bird
	Peter saw a hawk, Peter saw a vulture	⇒ Peter saw birds
Construction	Pete laid foundations, built walls, built a roof...	⇒ Pete built a house

Table 7.2: Example of macro-rules (from Mani [2001]).

1. Represent the meaning of the sentences with atomic propositional predicates. For example, the sentence "The student took his handout" would be represented with the atomic predicates

$$\exists x : student(x) \wedge previouslyMentioned(x) \wedge \exists y : took(x,y) \wedge handout(y) \wedge owns(x,y)$$

2. The macro-rules are used to replace a sequence of atomic prepositions by a simpler representation, with information loss. These rules are:

   - **Deletion**: delete a preposition.
   - **Generalisation**: Replace a proposition by other entailed by it.
   - **Construction**: replace a sequence of propositions by other that is entailed by all of them.

3. Text generation from the final representation.

Table 7.2 shows some examples of the three operations that can be performed with macro-rules. *Deletion* is easy to perform, as it can consist in selectively removing modifiers and adjuncts by using a measure of relevancy. *Generalisation* is somewhat more complicated, because there must be some criterion to decide when the propositions are too general, i.e. when to stop generalising. Finally, *construction* is the most complex of them all, as it will most probably require world knowledge to make inferences.

As a final remark, as Mani notes, lexical ontologies such as WordNet can also be very useful for automatic abstract generation. The following are some of their uses:

- Identification of which concepts are relevant for the topic of the document. For example, Reimer and Hahn [1988] consider that a concept in a lexical ontology is relevant if the ratio between the total number of instances of the concept and the number of instances that appear in the text (called *active instances*) is less than the number of its active instances.

- Hovy and Lin [1999] describe a similar idea called **Concept Wavefront**. Here, for every word in a document, the weight of all the synsets that contain that word and all their hyperonyms is incremented. Therefore, the final weight for a concept will be

$$W(C) = freq(C) + \sum_{C_i \in child(C)} W(C_i)$$

  Next, Hovy and Lin define a **fuser concept** as a concept whose children all contribute equally to its weight; and provide a method to find the fuser concepts for a given document. The set of fuser

concepts for a given document is called its **Concept wavefront**, and it is used to extract relevant sentences from texts.

7.4 Multi-document summarisation

Multi-document summarisation (MDS) is the extension of single document summarisation systems to multiple sources. It is a technology that has partly arisen as a result of the large amounts of data available on-line [Mani, 2001]. When the summary is extracted from multiple sources, new problems arise, the first of which is the fact the information can possibly be repeated across the documents. It will be therefore necessary to take into account the similarities and differences in the contents of the textual sources. Multi-document summarisation is a recently studied field, and there is not any consolidated approach yet, but a multitude of independent procedures; the aim of this section is to provided an overview of the problems that are encountered when addressing this task.

Cross-document coreference is the problem of deciding whether two expressions from different documents refer to the same entity or not. Single-document coreference is itself a hard problem: in the Seventh Message Understanding Conferences [MUC7, 1998] the best accuracy score was 61.8%, and most systems ranged from 30 to 60 percent. Cross-document coreference adds to the complexity of the task the fact that definite NPs such as *the earthquake* in two different documents are less likely to be referring to the same event, unless the two are dealing with the same natural disaster; or proper nouns such as *Mr. Smith* are less likely to refer to the same entity than in a single document. Coreference also includes linking pronouns with their antecedents, and solving relative temporal expressions, such as *today* or *three days ago*.

Redundancy detection: when summarising documents, they can possibly provide different views of the same topic. For example, a document collection about a car crash may consists of several journalists' report of the accident, the interview of an eye witness, and the point of view of a police agent and one of the the drivers. They will probably include some overlapping information, hence the MDS system must be able to recognise it so that the summary is not redundant. The following definitions describe two simple relationships between units of information:

Definition 1. Two text units are called *semantically equivalent* if they have exactly the same meaning (they can be string-identical, or one can be a paraphrase of the other).

Definition 2. A text unit A *informationally subsumes* a text unit B if the information in B is contained in A.

Therefore, if two units are semantically equivalent, one can be discarded from the summary. Also, if A informationally subsumes B, then B can be discarded. However, in most of the cases we shall find that pairs of units A and B from separate documents can have many different relationships between them. For example, they can have some information in common, but at the same time they both can contain some information that is not present in the other. In these cases, it may be necessary to merge both sentences in one. Table 7.3 shows several possible relationships between units from different documents.

The problem of finding redundancy and differences between the documents can be divided into two steps: in the first one, all documents are divided into topics and subtopics. For example, a medical text can be classified under the topic of a particular *disease*, and the separate paragraphs under subtopics *definitions*,

Relationship type	Description
Identity	The same text appears in more than one location
Equivalence	Two text spans have the same information content
Translation	Same information content in different languages
Subsumption	One sentence contains more information than another
Contradiction	Conflicting information
Historical background	Information that puts current information in context
Cross-reference	The same entity is mentioned
Citation	One sentence cites another document
Modality	Qualified version of a sentence, e.g. with quantifiers such as "reportedly" or "alleged"
Attribution	One sentence repeats the information of another while adding an attribution, e.g. "announced that"
Summary	Similar to summary in RST: one textual unit summarises another
Follow-up	Additional information which reflects facts that have happened since the last account.
Elaboration	Additional information that wasn't included in the last account
Indirect Speech	Shift from direct to indirect speech of vice versa
Refinement	Additional information that is more specific than the one previously included
Agreement	One source expresses agreement with the other
Judgement	A qualified account of a fact
Fulfilment	A prediction turned true
Description	Insertion of a description
Reader profile	Style and background-specific change
Contrast	Contrasting two accounts of facts
Parallel	Comparing two accounts of facts
Generalisation	Generalisation
Change of Perspective	The same source presents a fact in a different light

Table 7.3: Types of relationships across documents (from Mani [2001], from Radev [2000]).

causes, *treatment*, etc. [Hahn, 1990]. Therefore, only the sections from the different documents that refer to the same subtopic of the same topic have to be compared.

A topic segmentation was the approach taken by Kan et al. [2001]. However, this usually requires that the text referring to a given topic be formed with consecutive sentences. On the other hand, Nomoto [2001] used X-means [Pelleg and Moore, 2000] to cluster the sentences from the documents depending on the terms they contain. With this approach, sentences from all through the document may come together in the same cluster.

Information quality: not all sources are reliable; specially if the source of the documents is not known (as it is the case sometimes with documents downloaded from Internet). In this case, it is possible to find contradictions in different documents, and it would be desirable that the summarisation system discriminated between the different statements to keep just the true ones.

Temporal ordering: as pointed above, the documents can contain the same information, but from different points of view. After locating which phrases and which events are coreferring, it is important to be able to reconstruct the temporal ordering of the events told [Filatova and Hovy, 2001], something that has been implemented in summarisation systems, among others, by Lin and Hovy [2001] and Marcu [2001]. Refer to Section 6.3 (*Identification of time expressions and events*) for a discussion about events' ordering.

With respect to the methods applied to Multi-document summarisation, most of them are an adaptation of the methods already described for extract and abstract generation to the problem of having several documents.

- Firstly, some information is selected from each document separately. This can be done with a sentence extraction procedure, using extract-generation techniques, or with IE templates [Harabagiu and Maiorano, 2002].
- Next, multi-document coreference is performed to find coreferring Noun Phrases.
- The information coming from different documents is then analysed to discover the overlapping in Information Content. Sentences or templates that are subsumed by others are discarded; if several sentences refer to different views of the same event, then they may be merged or generalised.
- If necessary, a temporal ordering of the events is also performed (e.g. for a biography generator).
- Finally, the output is produced with a text generation module.

7.5 Summarisation applied to hypermedia generation

Nowadays, IR systems usually return a ranked list of documents that are judged to be similar to the user's query. Although alternative user interfaces have been proposed (e.g. Pirolli et al. [1996]), NLP techniques have been rarely applied to them. Recently, however, there have been several proposals for summarisation systems integrated with IR engines such as web search engines, in order to facilitate the search of information in large repositories. The contribution of automatic summarisation to IR is twofold:

- It can provide an indicate summary of the hits, highlighting the differences between them, and hence it can help the user in choosing between the documents selected by the engine [Maña-López et al., 1998].
- It can generate an informative multi-document summary of all the hits, so the users can directly read an overview of information related to their query, without the need to read the original documents.

The above-mentioned possibilities have been implemented in several general search engines for the Internet, and in specific IR products such as recommender systems. Currently, there are several approaches to multi-document summarisation of search engine hit lists [Radev and Fan, 2000, Radev et al., 2001]. Kan et al. [2001] describes the prototype of a system that addresses both aspects: it can generate an informative summary of all the retrieved pages together; or it can generate an indicative summary with the differences between them. The user can then choose to retrieve the full text of one of the pages.

These summarisation systems need to have the following characteristics: firstly, they have to be able to provide **user-focused summaries** (depending on the user query), instead of general-purpose summaries. Secondly, they have to handle **very different documents**, with sizes ranging from a few bytes to hundreds of kilobytes; and **different media types** (texts, images, tables, etc). And, ideally, multi-language summarisers would also allow the user to browse retrieved documents written in several languages at the same time [Lenci et al., 2002].

It seems that the current trend in the hypermedia IR technology will probably join technologies from IR, QA and Automatic Text Summarisation in producing hypermedia search engines that retrieve information from user questions written in natural language [Carbonell et al., 2001].

7.6 Summary and discussion

There are two main families of automatic summarisers: extraction and abstraction systems. **Extraction** consists in selecting portions of the text, such as relevant sentences or words. Simple summarisers can be built in this way, but there are some problems that are difficult to solve with these systems, such as maintaining the coherence of the generated texts, and dangling anaphoras. **Abstraction** consists in generating a summary that provides some information that was not present in the original texts, such as generalisations or inferences. It is a more complicated task that usually involves some kind of semantic processing of the source texts.

Multi-document summarisation poses new problems that haven't been completely solved. For most of the steps required to perform MDS there are techniques available, but they still have to be improved. Multidocument coreference resolution is still an arduous task, as well as the identification of overlapping information between different texts. For this last task, improving the Word Sense Disambiguation procedures can also prove very useful, by mapping synonym words into the same concept, and homonym words with a different meaning into different concepts.

Automatic Document Summarisation is a complex problem. However, recent advances show that it is possible to produce practical applications with the current technology. A common problem in most Natural Language Processing tasks is that, in order to work as well as a human, it probably requires an encoding of world knowledge and "common sense" with which it can reason and make inferences. However, purely corpus-based and statistical methods provide good results for generating extracts, and possibly good abstract generators will appear in the near future.

There are many applications for which it is not necessary that the generated summaries have a high degree of readability and cohesion; for some purposes, such as an email summariser, it is sufficient to present the most relevant sentences, regardless of the coherence between them. For these applications, text summarisation can already provide useful solutions.

Chapter 8

Adaptive Generation of Contents

This chapter describes the components of WELKIN which generate hypermedia pages when the visitors access it in search for information. It must be taken into account that the needs of the users, even though they may be accessing the same documental source, can be very different. For instance, different users might need the same pieces of information, but provided in unlike ways, such as in different languages or with different writing styles (e.g. adapted for a child or for an adult). It might also be the case that two users need different information about the same topic, either because they have different backgrounds and lack the knowledge required to understand some parts of the documents; or simply because they have different interests.

When designing an adaptive hypermedia system it is important to define, at the very beginning, the characteristics of the users that will be modelled, and how these characteristics will affect the presentation of contents. The structure of this chapter is the following:

- Section 8.1 describes the user characteristics that are taken into account and stored in their profiles, and how each characteristic is entered into the system.
- Next, Section 8.2 describes the algorithms that produce user-adapted summarised output, according to their specific profiles.
- Section 8.3 describes the user interface, and the adaptation techniques that are used for it.
- Finally, Section 8.4 shows how the original information from the documents can be extended by gathering additional information from the Internet.

The final section concerning the gathering of new information from the Internet is really performed during the off-line phase of the system, right after the documental database has been generated. However, its description has been placed in this chapter, and not in Part II, for two reasons. Firstly, it is an optional step: it is possible to skip it when generating the site from a set of documents; secondly, the topics discussed in that section, summarisation and presentation of information, are much more related to the contents of this chapter than to the remaining off-line processing.

For clarity, all the examples shown in this chapter have been taken from an example on-line information system built from Charles Darwin's *The voyages of the Beagle*, about Darwin's voyages in the Beagle to study animals and plants from around the world.

8.1 User profiles

The system described in this work addresses the case in which there is a document or a set of documents, and several users access it with different information needs. In that situation, the characteristics of the user that most influence the selection of contents are (a) the interests of the users, and (b) the amount of information that they are willing to read. In order to capture these two facts, the following user characteristics have been modelled:

1. **The user's interests** about different pieces of the original documents. This may be obtained, when the user registers into the system for the first time, in two ways:

   - By choosing a stereotype from a short list of general interests. If any of the predefined stereotypes fits the needs of a user, then it is possible to choose one, or a combination of several of them.
   - By choosing, from a list of very specific paragraphs, which ones are considered to be relevant. The relevancy of the information that will be provided in the site will be evaluated using this information.

   In any case, these choices are only used for initialising the interest profile; it will be always possible for the users to modify this profile dynamically, while browsing with the system, by indicating whether a paragraph found is relevant or irrelevant to them.

2. The amount of information that the users are willing to read in the whole site. This can be specified in several ways:

   - By indicating, directly, the compression rate that has to be applied (from 1 to 100 percent).
   - By indicating the total number of words that must be present in the target web site.
   - By indicating the total amount of time that the user wants to spend browsing the site.

   For each of the options, a different processing will have to be done in order to decide the amount of compression that has to be performed to the texts.

8.1.1 User interests

There are, possibly, different kinds of stereotypes that are more appropriate for each kind of document. For example, if we study Charles Darwin's *The Voyages of the Beagle*, we find that it is a multi-disciplinary document in which he describes both animals and places, and he also narrates the adventures he lived, such as volcanoes in action, some campaigns of the Argentinian General Rosas, and other adventures with natives from different parts of the world. It seems natural to pre-define three possible user interests for this text: **biology**, which includes all the descriptions of animals, animal behaviour, and plants; **geography**, which includes all the descriptions of places; and **history**, which includes the fragments in which Darwin explains other events. It is also possible that a user is interested in two of these topics at the same time, or in all of them, in which case the whole text is considered relevant.

On the other hand, when processing a different kind of text, such as a text about philosophy, different interests may arise. It may be also possible to extract information about **history** and **geography**, but there will also be paragraphs about **philosophy** and, possibly, about **politics**. And the scenario may change completely if we process and entirely different kind of text, such as a Computer Science textbook. In this

particular case, it might be useful to define different programming languages and different operating systems as possible topics for the user to choose.

Finally, it may be the case that a user is interested in some particular topic that had not been thought of beforehand by the site designer. Therefore, it is necessary that the way in which the user's interests is encoded allows for the definition of new possible topics.

This section describes the design decisions that were taken in order to tailor the contents of the generated hypermedia pages to the user interests. The following hypotheses were made at this point:

- Only very rarely there are topic shifts inside a paragraph; instead, a topic shift is usually accompanied by a paragraph change. In fact, it was observed that, quite often, when the topic changes, for example, from biology to history, even the writing style changes from scientific to narrative style. This implies that whole paragraphs can be taken as the unit of information for the topic filtering are whole paragraphs.

- Different topics can be represented with the sets of words that appear more frequently in the texts that verse upon them. Therefore, it is possible to use, again, a Distributional Semantics approach to perform the topic filtering of the text.

WELKIN allows the designer of the web site to create as many topics as wanted, and then to train the system in a supervised way, so these topics can be used to decide which of the textual fragments are useful for each particular user. These interest profiles shall be used to select only the information from the documents that is not considered relevant for the user, in a process that can be called *topic filtering*.

Once these hypotheses have been made, topic filtering can be implemented with topic signatures (see Section 4.2.1, *Metrics*). For the site about Darwin's *The Voyages of the Beagle*, three different stereotypes were considered: biology, geography and history. Therefore, the topic classifier should decide, for each paragraph, whether it is relevant about each of these three interests.

For training, the first one hundred paragraphs from *The Voyages of the Beagle* were classified by hand in one of the three stereotypes: biology, geography or history. Next, a topic signature was created for each of the stereotypes, by collecting uninflected words from the paragraphs. Finally, for each of the three signatures, the other two were used as a contrast set for changing the frequencies into weights, using the χ^2 function. An additional experiment was performed, about the convenience of generalising the words from the topic signatures using the WordNet ontology. The results and suitability of this method for topic filtering are explained in this section.

In the first experiment, the uninflected forms of all the open-class words (nouns, verbs, adjectives and adverbs) were collected and counted from the paragraphs corresponding to each topic, and the χ^2 function was used to calculate the weight of each word in each of the signatures. Tables 8.1 and 8.2 (left) show the words that were found with higher frequencies. As can be seen, in the narrative texts, words that receive high weight are verbs of action and movement (e.g. *do*, *pass* or *leave*), characters (e.g. *man* and *horse*) and places where the events happen (e.g. *road* and *house*). The descriptive paragraphs contain words to describe the location types (e.g. *island*, *river*, *country* or *cover*) and words referring to distances (e.g. *distance*, *foot* or *mile*). Finally, texts about biology contain words relative to animals (e.g. *animal*, *bird*, *species*, *egg* or *nest*).

In a second experiment, instead of collecting words, the topic signatures were produced by collecting WordNet synsets and next generalising. Previous work shows that it is possible to improve the classification results by using WordNet synsets instead of lexical forms of words in order to create the topic signatures

Narrative texts			Descriptive texts		
Word	**Frequency**	**Weight**	**Word**	**Frequency**	**Weight**
be	354	0	be	380	0
have	87	0	have	88	0
man	33	1.884421	country	28	0.3074589
day	29	0.49710438	part	27	0.4783065
horse	28	1.0849504	tree	27	1.1606017
country	26	0.2868369	plain	24	0.85766035
do	26	0.90880805	mile	22	0.9226785
time	21	0.16649379	see	22	0
pass	19	0.7313451	cover	21	1.7586255
make	17	0.10655532	water	20	0.015222555
know	16	0.63887554	day	19	0
give	16	0.5023026	island	19	1.7672732
house	15	1.1208979	salt	19	1.675031
find	15	0	rock	18	1.4593027
see	15	0	degree	18	0.85947263
part	15	0	form	18	0.29217592
say	15	0.41393185	foot	18	0.62209314
road	14	2.1813467	distance	16	0.51585084
take	13	0.41393185	river	16	1.4361887
leave	13	0.7466217	round	15	0.945477

Table 8.1: Top-frequency words in the signatures for the narrative and descriptive texts.

Biological texts			Biological texts (synsets)		
Word	**Freq**	**Weight**	**Synset**	**Freq**	**Weight**
be	838	0.31686366	artifact	2301	0.072262526
have	226	0.3941432	social relation	1334	0.116964035
animal	90	1.1195472	communicate	1244	0.05366011
bird	76	1.2105474	person, human	760	0.028999895
find	69	0.78169715	substance, matter	717	0.21241684
see	64	0.58891094	property	689	0.17070201
water	55	0.6657796	mental object	558	0.07125762
species	53	1.2986525	body part	533	0.3877598
number	46	0.87160313	be	531	0.08280898
time	45	0.41963083	be	531	0.08280898
habit	41	1.2062719	fundamental quantity	503	0
make	37	0.3596968	natural object	465	0.06322681
day	37	0.10375394	quality	449	0.115066975
appear	35	0.74276936	change	427	0
nest	34	1.5415385	animal	383	0.425628
country	34	0	region	347	0
part	34	0.13700412	organisation	285	0.06364893
egg	33	1.4689231	shape, form	274	0.0872535
form	30	0.29723078	achievement	274	0.050232295
place	30	0.5307324	region, part	257	0.021937495

Table 8.2: Top-frequency words in the signatures for the biological texts.

1. For every open-class word in the document,
   (a) Get all the synsets that contain that word.
   (b) Get all the hyperonyms of those synsets that are grandchildren of the root of the hierarchy to which they belong.
   (c) Increment the frequency of all those synsets.

Figure 8.1: Algorithm for generalising the topic signatures using WordNet.

[Magnini and Strapparava, 2001, Gonzalo et al., 1998]. For every word in the paragraph the frequency of every synset containing that word was incremented. For instance, if the word *bank* was found in a text, the frequency of the ten synsets in WordNet 1.7 that contain it would be incremented. Among them, for example, there is a synset that refers to the financial institution, and other that refers to the sloping land besides a river.

The generalisation was done in a simple way. Only the synsets in WordNet that are located two levels below the root of the hierarchies, which refer to general concepts, were considered. These synsets include concepts such as *animal*, *person*, *stream*, *sea*, etc. The algorithm that collects the word frequencies was implemented in the following way: for every word in the text, the frequency of any of these synsets was incremented if there exists a hyponym which contains that word. So, for example, for every kind of animal that is mentioned in a paragraph, the synset *animal* is incremented in the signature; therefore, if an unknown text contains a reference to an animal that did not appear in the training texts anyway it can be considered as a good indication that the text deals with biology. The algorithm is shown in Figure 8.1.

In this way, every reference to any person, any animal or any river will increment the synset *person*, the synset *animal* or the synset *river*, respectively.

As already mentioned, the signatures, either of words or of synsets, were produced from the first one hundred paragraphs from *The Voyages of the Beagle*, annotated with the topic label. In order to measure the accuracy of the classification algorithm, the test set consisted of the following 58 paragraphs, which had also been annotated by hand. For each of the paragraphs in the test set, a signature is built with its words (or synsets) and their frequencies, and then a similarity metric for each topic is calculated as the scalar product of the topic signature and the paragraph signature. Finally, these similarities are normalised so they add up 1.

The results have been calculated in two ways: the strict accuracy is the number of times that the topic with the highest similarity to the paragraph is the correct one; and the >33.33% accuracy is calculated in the following way: if the correct topic of a paragraph attained a similarity higher or equal than 33.33%, then the classification is considered correct. In other words, if the classification of the correct topic received a similarity higher than the uniform distribution, then the classification is considered valid.

The results are shown in Table 8.3. As can be seen, results are much better using words than synsets. This may indicate that, by choosing the WordNet synsets that are located two levels below the root, the algorithm is not generalising but losing important information. Nevertheless, this result does not imply that generalising WordNet synsets is bad; on the other hand, it may be an indication that it has been performed in a coarse way.

There is still room for future research; for example, using a WSD procedure could help improve the

Method	Total	Correct	Accuracy
words	58	46	79.31%
words > 33.33%	58	53	91.38%
synsets	58	40	68.97%
synsets > 33.33%	58	45	77.59%

Table 8.3: Results classifying the paragraphs in one of the three topics: biology, descriptive and narrative.

generalisation of the synsets. A better selection of the synsets which should be generalised, such as the one described by Li and Abe [1997], may produce good results. Hearst and Schutze [1993] also describes a different way of choosing a set of categories amongst the WordNet synsets.

8.1.2 Topic filtering

The aim of this component is to filter out all the information that might not be relevant for the user. Multidisciplinary texts usually contain disquisitions about different subjects, some of which might not be interesting for the user of a hypermedia site. There are automatic algorithms to partition full-length expository text into a sequence of subtopical discussions [Hearst, 1993], but it is also possible to partition the text in paragraphs, and to classify the paragraphs separately, considering that it is not frequent to find a topic shift in the middle of a paragraph.

There are several ways to label portions of texts according to the topic that they address. Jacobs and Rau [1990] describe an algorithm that starts with a pre-defined set of topics and a list of words that are relevant for each of them, and uses the lists of words to classify the text fragments. The approach followed by Masand et al. [1992] is more similar to the one used in this work: it is a supervised algorithm for which there is, initially, a set of text fragments, each one labelled with its topic, and the algorithm learns the lexical items that are useful for classifying new texts. In a possible third approach, the topics themselves could be automatically induced from the set of texts, e.g. by clustering them, or using a conceptual ontology. Hearst and Schutze [1993] describes an unsupervised system that automatically converts WordNet in a set of flat topics, and uses it to classify unknown fragments of texts and unknown words into these topics.

When new users log in, the system generates a login page containing the form in Figure 8.2. Users can select one or several stereotypes, or state that they want to define their own personal profile. In the first case, the pre-defined signatures are copied into the new profile. In the second case, the system generates a list of 100 specific topics, obtained from the first one hundred paragraphs in the domain-specific texts from which the site was generated. The new users have to state which of those paragraphs they consider interesting. Two new topic signatures are calculated from that information: one corresponding to the relevant paragraphs, and the other one to the irrelevant, and these are included in the new user's profile. *In any case, these preferences are no static*: users will be able to change the preferences while browsing, by indicating whether any paragraph in the site is very relevant or irrelevant to them.

Therefore, the user's profile contains the information necessary to perform the topic filtering when browsing the site. For each paragraph that belongs to a hyperpage that has to be generated,

1. The paragraph's signature is formed by collecting all the open-class words and their frequencies.

2. The dot product is performed between the paragraph's signature and each of the three topic's signatures, and the three similarity metrics are normalised.

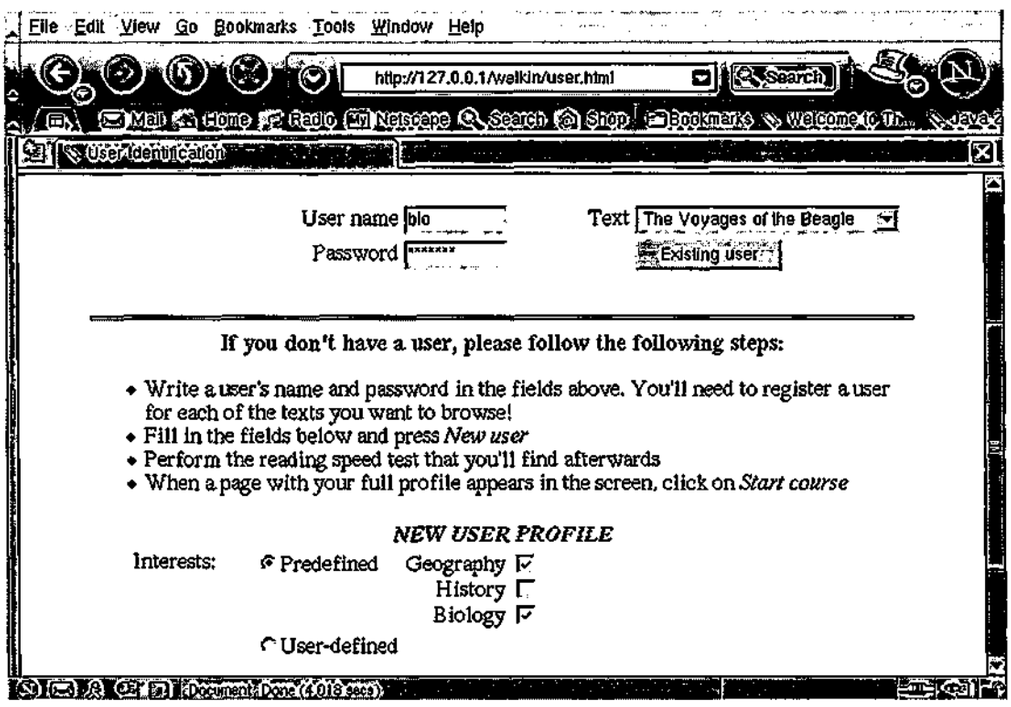


Figure 8.2: Form with which the user can select one or several predefined stereotypes, or state that he wants to define his own profile.

3. The paragraph is chosen if, for any of the topics that the user has selected as relevant, the similarity metric is above $\frac{1}{N}$, where N is the number of topics available.

For instance, with the example described above where there are three stereotypes (biology, history and geography), a paragraph is chosen if its similarity to any of the stereotypes selected by the user is greater or equal than 33.33%. For example, if a paragraph has the following topic similarities:

Biology	Description	Narrative
0.35	0.40	0.25

then the paragraph will be ruled out if the user has only selected the *history* checkbox, which refers to the narrative paragraphs. If any of the checkboxes labelled *biology* or *geography* is selected, the paragraph will be chosen for presentation.

Note that, with this algorithm, the same paragraph may be considered relevant for different stereotypes, if it contains keywords relevant to more than one topic. The limit situation would be when a paragraph scores exactly 33.33% for the three topic signatures, in which it case it would be shown to every user that follows any of the stereotypes.

For a non-stereotypical user, when generating any section, every paragraph is compared with the relevant and the non-relevant signatures, in the same way as before. If the similarity to the relevant signature is higher or equal than 50%, then the paragraph is chosen as relevant; otherwise, the paragraph is filtered out.

As an example, it is possible to read in Appendix C some examples of texts generated, from the same original document, to three users interested in different stereotypes. Figure C.1 shows the three paragraphs

Figure 8.3: Form with which an already existing user can log in, or a new user can create a profile.

from the first section of the book that were shown to a user interested in history. The generated section shown to a user with the biology stereotype contains 5 paragraphs, shown in Figures C.2 and C.3; and the section shown to a user interested in geography (Figures C.4 and C.5) shares the first paragraph with the extract of history, and also contains 6 other paragraphs with descriptions of villages and the geological formation of the island.

8.1.3 Available time and reading speed

As already mentioned, one of the aims of the adaptive site is to provide the quantity of information that the users are willing to read. This shall be done with an automatic summarisation system, and there are three possible ways to learn the compression rate that will be performed to the texts, as shown in Figure 8.3:

1. Asking, directly, the compression rate (from 1 to 100 percent).
2. Asking the total number of words that will be present in the target web site.
3. Asking the amount of time that the user wants to spend browsing the site.

The first situation is the simplest one, when the compression rate is directly provided by the user. In the second case, the total number of words that the user is willing to read is divided by the number of words summed up from all the relevant paragraphs of the web site, according to the user profile. The compression rate is the result obtained.

The last case is the one that requires a little more work. In order to know which is the compression rate to be performed to the documents, three values are needed:

- The total amount of information in the site that is relevant for the user, calculated with the profile of interests.
- The total amount of time that is available to the user.
- The quantity of information that the user is able to read for each time unit.

The users provide their **availability of time** as a number of hours that they intend to browse the site. Next, the user **reading efficiency** is collected with a simple test, in the following way: when registering into the system for the first time, the site server asks the user to read a small passage of text. The reader has to press a button before starting reading, and press a different button after finishing the passage. In this way, the *reading speed* can be calculated, measured in number of words per minute. The form is shown in Figure 8.4.

It is not enough to have the user's reading speed, because it may be the case that the user has read it very quickly and has not assimilated the contents of the text. **Reading efficiency** is defined as the product of the reading speed and the reading comprehension [Jackson and McClelland, 1979], which can be calculated by asking the user to answer ten questions about the passage that has just been read. The form with the questions is shown on Figure 8.5. The percentage of questions that are correctly answered will be used to weight the reading speed. If this percentage is low, the user should read slower in order to better understand and retain the information.

This method is very similar to the one used in reading proficiency tests such as the reading test in the Stanford Achievement Test, 9th edition (SAT-9), the Gray Oral Reading, 4rd Edition (GORT-4), or the *reading fluency* test of the Woodstock-Johnson III Test of Achievement (WJ-III). There exist other procedures for measuring reading comprehension, such verbally supplying the missing word from each sentence or brief paragraph (used for the *Passage Comprehension* test of WJ-III), but they were not judged convenient here because, if there are missing words, the measure of reading speed is not the same.

It is worth to note that, although the availability of time and the reading efficiency are considered static characteristics of the user, the preferences may change dynamically while browsing, if the user shows his preferences while reading the site. Even in the case in which the user has initialised the profile with stereotypes, it is possible to indicate that a certain paragraph is very relevant or, on the contrary, that it is uninteresting, and the interests profile will be updated automatically. This may also produce a change in the compression rate applied, if the total amount of relevant information varies.

8.2 An algorithm for adaptively summarising the text contents

With the information provided by the users at registration time it is possible for the system to calculate the compression rate that has to be performed to the text. This section explains how it is calculated, and the algorithm that has been used for performing the data compression.

The compression rate is the ratio between the number of words in the paragraphs of interest and the target number of words in the site. For example, if a user has a *reading efficiency* of 150 words per minute

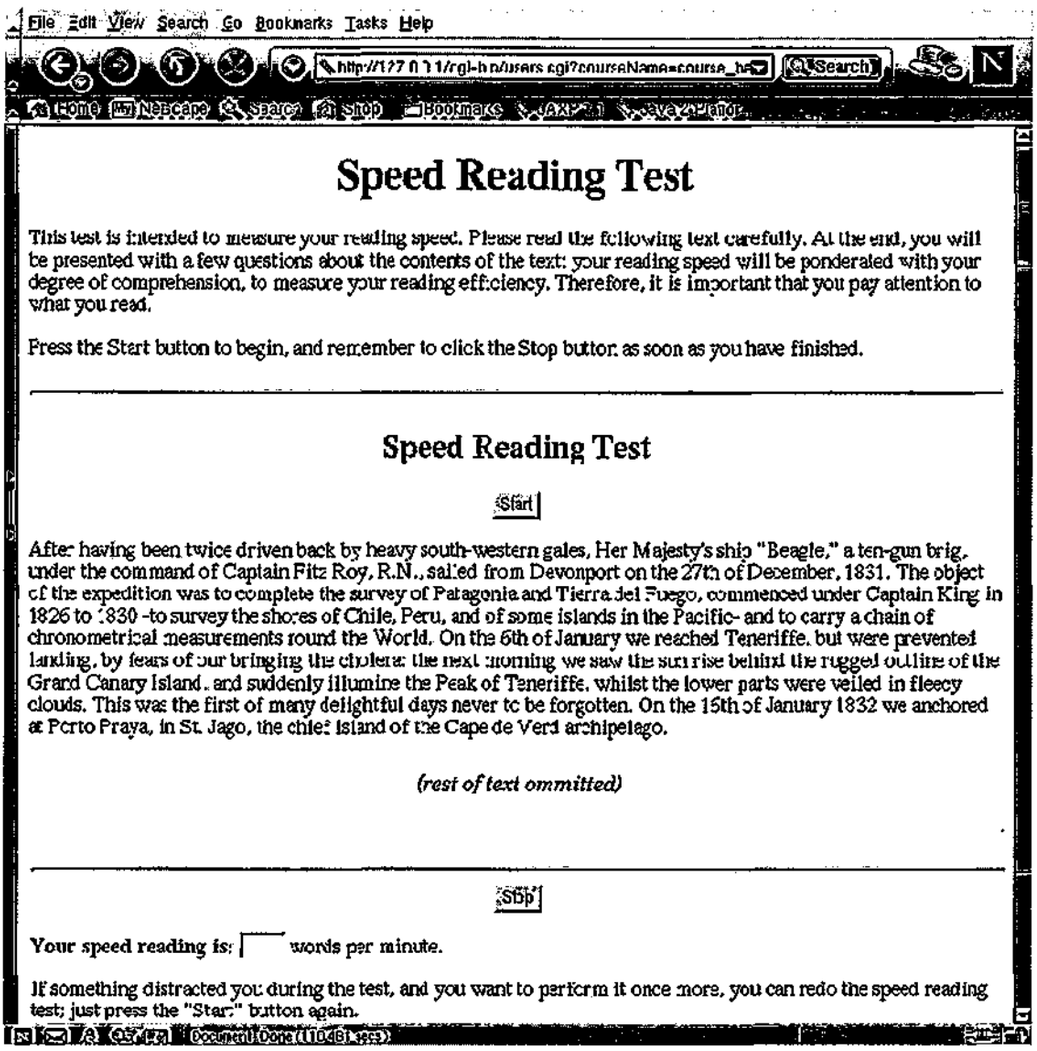


Figure 8.4: Reading speed test. The text to be read is sensibly longer than that shown in the image.

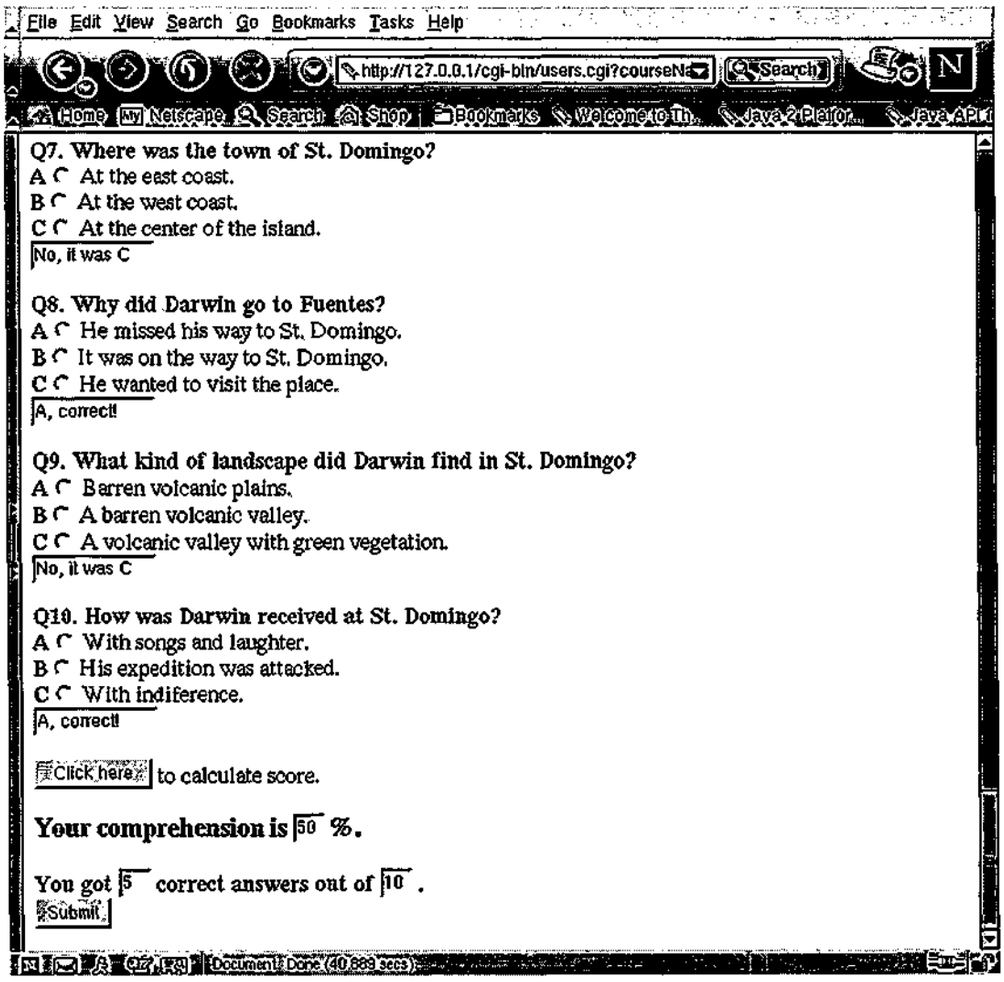


Figure 8.5: Reading comprehension test. There is a total of ten questions about the passage, and the percentage of correct answers is the comprehension rate.

16	47	60	68	75	78	103
8	47	49	53	80	86	98
15	21	27	75	81	93	118
42	46	53	83	84	103	113
15	29	44	47	62	112	115
12	20	26	43	51	55	69
11	39	45	50	64	86	92
5	24	37	58	83	112	116
3	33	39	52	53	81	102
6	33	34	53	61	107	118

Figure 8.6: Initial population of summaries. Each line in the figure is the genotype of a summary, which contains the numbers of the sentences that are selected for the summary.

and plans to read the whole text in two hours, then the number of words that he or she can read is

$$150\frac{words}{minute} \times 120\ minutes = 18,000\ words$$

If there are 50,000 words in paragraphs that are relevant, the compression rate is $\frac{18,000}{50,000} = 36\%$.

In order to generate the user-dependent summaries, a *sentence extraction* procedure has been used (see Section 7.2, *Text extraction*). Therefore, the unit of information considered is a whole sentence. When a user wants to read a document section, the following procedure is followed:

1. Perform the topic filtering to remove the paragraphs that are not of interest.
2. From the remaining paragraphs, count the number of sentences and multiply by the compression rate. That is the number of sentences in the target summary.
3. Select exactly that number of sentences from the section.

The selection of the sentences that will be in the extract is done using a new summarisation procedure that is based on genetic programming [Holland, 1975]. The algorithm starts with a set of random individuals, each of which is a vector of integers that represent the sentences that are chosen to create an extract. For example, if there are 121 sentences and the summary must contain only seven sentences, then a possible initial set of ten random individuals is the one in Figure 8.6.

For each summary, a fitness function has been defined based upon some heuristics. The following are some characteristics of summaries that had been already observed when designing different summarisation algorithms:

- Summaries that contain long sentences are better summaries than summaries that contain short sentences [Marcu and Gerber, 2001]. A partial fitness function can be defined as the sum of the lengths of all the sentences in the extract (measured in number of words):

$$L(S) = \sum_{i=0}^{N} length(s_i) \tag{8.1}$$

- Summaries that contain contain sentences that occur in the beginning of a paragraph in the original documents are better than summaries that contain sentences that occur toward the end [Hovy and

Lin, 1999, Mani, 2001]:

$$W(\mathcal{S}) = \sum_{i=0}^{N} 1/position(s_i) \tag{8.2}$$

- Summaries that contain the sentences in the same order than in the original documents are better than otherwise [Marcu, 2001]:

$$O(\mathcal{S}) = \begin{cases} 1 & \text{if the sentences are ordered} \\ 0 & otherwise \end{cases} \tag{8.3}$$

- Summaries that contain sentences from all the paragraphs are better than summaries that focus only on a few paragraphs [Marcu, 2001].

$$C(\mathcal{S}) = |\{p : paragraph(p) \wedge (\exists s \epsilon \mathcal{S} : s \epsilon p)\}| \tag{8.4}$$

The following heuristics were also used for the fitness function, in order to be able to adapt to the user profile:

- Summaries that contain sentences with domain-specific terms are better that summaries that only contain general-purpose terms.

$$T(\mathcal{S}) = |\{t : domain\_specific(t) \wedge (\exists s \epsilon \mathcal{S} : t \epsilon s)\}| \tag{8.5}$$

- Summaries that contain sentences that are more relevant according to the user profile are better than summaries that don't.

$$P(\mathcal{S}) = \sum_{i=0}^{N} similarity(s_i, user\_profile) \tag{8.6}$$

- Summaries that contain sentences with any of the user's query keywords are better that summaries that only contain general-purpose terms.

$$Q(\mathcal{S}) = |\{t : query\_word(t) \wedge (\exists s \epsilon \mathcal{S} : t \epsilon s)\}| \tag{8.7}$$

- Summaries that contain complete sentences (with subject and verb) are better than summaries that contain any sentence with any of those constituents.

$$V(\mathcal{S}) = |\{s : s \epsilon \mathcal{S} \wedge has\_subject(s) \wedge has\_verb(s)\}| \tag{8.8}$$

In this approach, the final score of a summary has been calculated as a weighted sum of the different fitting functions, in a similar way as in the Edmundsonian paradigm (see Section 7.2.1):

$$\begin{aligned} F(\mathcal{S}) = & w_L \cdot L(\mathcal{S}) + w_W \cdot W(\mathcal{S}) + w_O \cdot O(\mathcal{S}) + w_C \cdot C(\mathcal{S}) + \\ & w_T \cdot T(\mathcal{S}) + w_P \cdot P(\mathcal{S}) + w_Q \cdot Q(\mathcal{S}) + w_V \cdot V(\mathcal{S}) \end{aligned} \tag{8.9}$$

Once the fitness function has been decided, the procedure followed is the standard for genetic algorithms: at every generation, the two less adapted individuals in the population die, and the two most adapted have

children. Population changes by means of the *mutation* operator, that changes a random sentence number in an individual, and the *crossover* operator, that interchanges a random portion of the genotype of two individuals. When the best summary does not vary for a certain number of steps, the evolution stops and the summary with the best fitness function will be presented to the user. Table 8.4 shows the population of summaries at different stages of evolution for a summary of 13 sentences from a total of 47.

The weights for each of the partial fitness functions were set by hand, after several experiments, to some values that seemed to produce the best results.

8.2.1 Evaluation

A test set of thirty documents was built, with lengths ranging from 6 to 59 sentences. This set consisted of three subsets:

1. A set with 10 documents for a user whose general interest is biology, needing a compression of roughly 29% (the summaries had to be 29% of the original text). Two of the summaries had an additional constraint: that the user wanted specifically to have information about a *carrancha* and about a *bizcacha*, respectively.

2. A set with 10 documents for a user interested in geography, with a compression rate of roughly 39%. Again, some summaries have additional keywords, such as *Siberia*.

3. A set with 10 documents for a user interested in history, with a compression rate of roughly 46%. As before, some summaries will be general, and others have to be focused on specific topics.

In order to create human-created summaries for testing purposes, nine human judges were asked to participate. All of them were graduates in Electrical Engineering, Computer Science or Mathematics. Each of the three sets of documents was given to three different human judges.

Every document carried an explanatory note indicating the preferences of the user, and the number of sentences that had to be selected. Figure 8.7 displays one of the documents that was provided to the judges. The agreement of every pair of judges was calculated for every set of ten documents, as shown in Table 8.5.

The creation of a summary is something that is not totally objective: different people will probably produce different summaries from the same text, even though they do not have to reformulate the information but only extract sentences. Therefore, for a proper evaluation of the system it is equally important to know how much humans agree on the same task. A widely used metric to measure judge agreement is the Kappa statistic [Carletta, 1996, Siegel and Castellan, 1988], defined as

$$K = \frac{P(A) - P(E)}{1 - P(E)} \tag{8.10}$$

where P(A) is the proportion of times that the judges agree, and P(E) is the proportion of times that we would expect them to agree by chance. If all the judges agree, $P(A)$ is 1 and the value of Kappa is 1; on the other hand, if the agreement of the judges is the one that could be expected by mere chance, Kappa takes the value of 0. According to Carletta [1996], citing other research works, a value of $K > .8$ can be considered good, and a value between 0.67 and .8 "allows tentative conclusions to be drawn". However, they say that in medical literature values of K between 0.21 and 0.4 are considered "fair".

The values of Kappa obtained from the judges' answers are listed in Table 8.6. $P(E)$ was calculated in the following way: with a compression rate of 0.46, given a sentence, the probability that three judges

Generation	Sentences	Fitness
	0 2 6 7 9 16 22 24 26 30 38 43 44	36.82621
	0 5 6 12 18 19 21 26 28 31 38 43 46	39.009567
	1 5 6 9 18 24 31 33 35 36 42 44 45	39.45049
	2 3 4 7 20 24 27 29 31 34 38 41 43	40.31381
1	2 5 6 8 17 20 25 27 29 32 34 43 44	41.124672
	7 14 15 22 26 29 33 35 37 39 40 43 44	41.633026
	3 4 10 11 21 24 28 29 38 41 43 44 45	42.431465
	7 8 12 14 15 16 24 25 28 29 33 39 40	42.67132
	9 12 15 16 19 22 24 26 31 35 36 42 43	45.15825
	1 6 8 13 20 28 29 33 35 41 42 44 46	46.501553
	1 6 7 8 9 13 20 28 31 41 42 44 46	45.05381
	9 12 15 16 19 22 24 26 31 35 36 41 43	45.1657
	6 13 20 22 28 29 30 33 35 41 42 44 46	45.533634
	1 6 13 14 20 22 28 29 33 35 41 42 46	45.969124
5	1 8 13 17 20 23 29 31 33 41 42 44 46	46.440636
	1 6 8 13 20 28 29 33 35 41 42 44 46	46.501553
	1 6 8 13 20 28 29 31 33 41 42 44 46	46.64182
	1 8 13 20 23 28 29 31 33 41 42 44 46	46.6646
	1 6 7 8 13 20 28 29 31 41 42 44 46	47.0431
	1 6 13 20 22 28 29 33 35 41 42 44 46	47.385387
	1 6 13 20 22 25 28 29 31 33 35 42 46	46.404633
	0 6 13 20 22 28 29 33 35 42 44 45 46	46.88444
	1 3 6 9 13 20 29 33 35 41 42 44 46	47.319347
	1 6 13 20 22 28 29 33 35 41 42 44 46	47.385387
10	1 3 6 10 20 22 29 33 35 41 42 44 46	47.395218
	1 3 6 13 16 20 29 33 35 41 42 44 46	47.523808
	6 13 20 22 28 29 33 35 41 42 44 45 46	47.60114
	1 6 13 20 22 25 28 29 33 35 41 42 46	47.65891
	1 3 6 13 20 22 29 33 35 41 42 44 46	48.86552
	1 3 4 6 13 22 29 33 35 41 42 44 46	49.599186
	1 3 4 6 11 13 26 29 33 35 41 42 44	49.89394
	0 3 4 6 13 16 22 29 33 35 41 42 44	50.32384
	1 3 4 6 13 22 26 29 33 41 42 44 46	50.415627
	3 4 6 10 13 22 26 29 33 35 41 42 44	50.504337
20	1 2 3 13 20 22 26 29 35 36 41 42 44	50.619522
	0 1 3 4 6 13 22 29 33 35 41 42 44	50.67549
	1 2 3 13 20 22 26 29 35 41 42 44 46	50.712723
	1 3 4 6 13 22 26 29 33 35 41 42 44	50.814087
	1 2 3 13 19 20 22 26 29 41 42 44 46	50.84548
	1 3 4 6 13 26 29 33 35 39 41 42 44	51.74695
	3 4 17 18 19 26 29 31 39 40 41 43 44	49.833973
	3 4 16 19 24 25 26 29 39 40 42 43 44	51.694916
	4 19 24 25 26 29 31 33 39 40 41 42 43	52.008553
	4 19 24 26 29 31 39 40 41 42 43 44	54.09825
50	3 4 19 26 29 31 39 40 41 42 43 44 46	54.138783
	3 4 17 19 26 29 31 39 40 41 42 43 44	54.306686
	3 4 17 19 26 29 31 39 40 41 42 43 44	54.306686
	3 4 17 19 26 29 31 39 40 41 42 43 44	54.306686
	3 4 19 24 25 26 29 31 39 40 41 42 43	54.371773
	3 4 19 24 25 26 29 39 40 41 42 43 44	54.43973

Table 8.4: Evolution of the population of summaries.

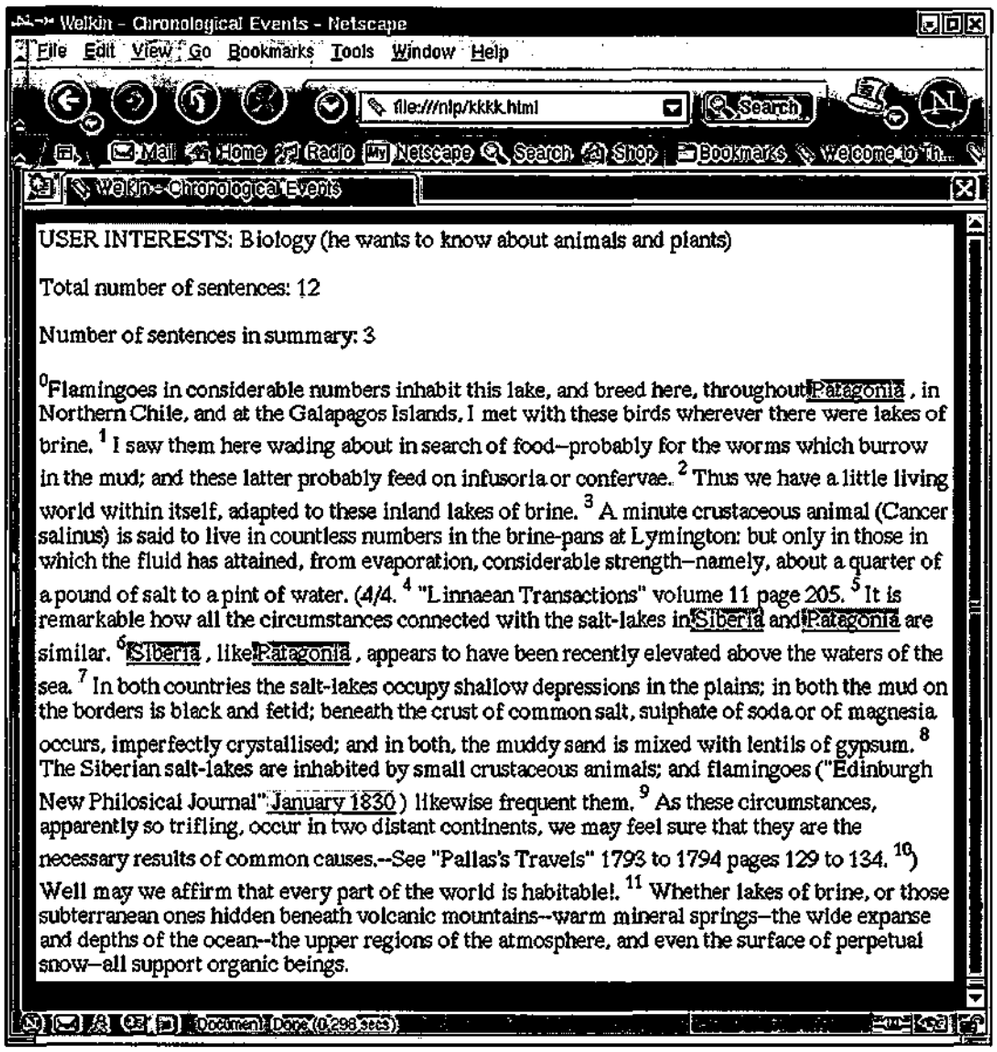


Figure 8.7: Text that was supplied to the judges to generate a summary. It included a short description of the user interests, the total number of sentences in the text, and the number of sentences that had to be selected. Each sentence was numbered in order to facilitate the annotation.

Compression rate	Judges pair	Agreement
0.29 (10 docs.)	1,2	63.11%
	1,3	62.14%
	2,3	62.14%
0.39 (10 docs.)	1,2	61.26%
	1,3	63.39%
	2,3	62.16%
0.46 (10 docs.)	1,2	65.57%
	1,3	72.95%
	2,3	72.13%

Table 8.5: Agreement between pairs of judges.

Compression rate	P(A)	P(E)	Kappa
0.29 (10 docs.)	0.46	0.024	0.447
0.39 (10 docs.)	0.43	0.059	0.435
0.46 (10 docs.)	0.59	0.097	0.546

Table 8.6: Level of agreement between judges.

Compression rate	Judges pair	Agreement
0.29 (10 docs.)	1,auto.	52.43%
	2,auto.	47.57%
	3,auto.	39.81%
	majo.,auto.	**49.51%**
0.39 (10 docs.)	1,auto.	54.05%
	2,auto.	54.46%
	3,auto.	51.79%
	majo.,auto.	**54.95%**
0.46 (10 docs.)	1,auto.	68.85%
	2,auto.	54.92%
	3,auto.	63.11%
	majo.,auto.	**67.21%**

Table 8.7: Agreement between each judge and the automatically generated summaries. The last line compares the summary formed with the sentences that received more votes from the judges against the automatic summary.

randomly choose it as summary-worthy is $0.46^3 = 0.097$. The values obtained show that there is some agreement amongst the judges, as the level of agreement was always substantially higher than the one that could be expected with random summaries, although they are below 0.67. In fact, as [Mani, 2001, pg. 226] notes, in the SUMMAC evaluation [Mani et al., 1998] four subjects were asked to evaluate some summaries, without explicit criteria. There was unanimous agreement only for 36% of the sentences, leading to a Kappa of 0.24.

The conclusions we can derive from the fact that inter-judge agreement is not very high is that the task was not defined in a very precise way. Indeed, many choices of sentences were left to the personal choice of the judges, as there were not very specific guidelines. In the case of summarisation, even if we know that the user is interested on a topic, such as *biology*, there might be many sentences referring to that topic, and different judges use their own criteria. In any case, the value of Kappa was always well above that of the SUMMAC competition.

After collecting the information from the judges, the *target summaries* used to evaluate the algorithm were calculated with the sentences that had received more votes from them. As shown in Table 8.7, the agreement between the genetic algorithm and the hand-made summaries was 49.51% for the 29% summaries, 54.95% for the 46% summaries, and 67.21% for the 46%. Considering that the agreement between human judges was between 60% and 70% (c.f. Table 8.5), the machine-generated summaries can be considered to be at worst around 15% less accurate than human-made extracts, and at best around 5% less accurate.

Figures 8.8, 8.9 and 8.10 show the result of applying this algorithm to the paragraphs in Figure C.1, with a compression rates of 22%, 33% and 45%, respectively. As can be seen, the first paragraph is considered the most informative, as most of its sentences are selected even in the smallest summary. The summary

History

After having been twice driven back by heavy south-western gales, Her Majesty's ship "Beagle," a ten-gun brig, under the command of Captain Fitz Roy, R.N., sailed from Devonport on the 27th of December, 1831. The object of the expedition was to complete the survey of Patagonia and Tierra del Fuego, commenced under Captain King in 1826 to 1830– to survey the shores of Chile, Peru, and of some islands in the Pacific– and to carry a chain of chronometrical measurements round the World. On the 6th of January we reached Teneriffe, but were prevented landing, by fears of our bringing the cholera: the next morning we saw the sun rise behind the rugged outline of the Grand Canary Island, and suddenly illumine the Peak of Teneriffe, whilst the lower parts were veiled in fleecy clouds.

One day, two of the officers and myself rode to Ribeira Grande, a village a few miles eastward of Porto Praya.

We returned to the Venda to eat our dinners.

Figure 8.8: Paragraphs shown to a user interested about history, with a compression rate of 22%. The uncompressed paragraphs are shown in Figure C.1.

in Figure 8.9 contains an inconsistency, as the definite description *the black priest* is cited before being introduced, because the sentence in the previous paragraph that introduces it as *a black Padre* was discarded.

8.3 User interface

Apart from presenting the information when the user makes a request, WELKIN has to provide the interface tools necessary to help the user navigate the data. Given that the user interface is built on top of web browsers, the navigation options are built with hyperlinks. The following actions can be performed by the user to navigate in the site:

- Browse the information **sequentially**: Each paragraph that is viewed separately contains a hyperlink to the whole section that contains it, and every section that is shown complete contains two hyperlinks to the previous and the following sections in the document.
- Retrieve the information relative to a **date**: dates were found in the texts during the linguistic processing, and each date is associated either to a sentence in the text, to a paragraph or to a whole section, if it appears in the section header. Dates are identified because the background colour is set to yellow, different from the background of the page (which is white). By clicking on a date with the mouse, the user is shown all the information relative to that date.
- Retrieve the information relative to a **term**: during the off-line processing of the text, the most frequent unknown terms were identified and classified in WordNet. The interface uses a background colour code to identify them. So, unknown words that were classified as people appear over a brownish background, and unknown words classified as locations appear over a green background. If the user clicks on any of these terms, all the information available about them appears.

At the left part of the browser there is a vertical frame that offers additional options for navigation. If the system observes that the user is choosing dates, then this frame will allow the user to navigate the text

History

After having been twice driven back by heavy south-western gales, Her Majesty's ship "Beagle," a ten-gun brig, under the command of Captain Fitz Roy, R.N., sailed from Devonport on the 27th of December, 1831. The object of the expedition was to complete the survey of Patagonia and Tierra del Fuego, commenced under Captain King in 1826 to 1830– to survey the shores of Chile, Peru, and of some islands in the Pacific– and to carry a chain of chronometrical measurements round the World. On the 6th of January we reached Teneriffe, but were prevented landing, by fears of our bringing the cholera: the next morning we saw the sun rise behind the rugged outline of the Grand Canary Island, and suddenly illumine the Peak of Teneriffe, whilst the lower parts were veiled in fleecy clouds.

One day, two of the officers and myself rode to Ribeira Grande, a village a few miles eastward of Porto Praya. In the course of an hour we arrived at Ribeira Grande, and were surprised at the sight of a large ruined fort and cathedral.

We returned to the Venda to eat our dinners. We presented the black priest with a few shillings, and the Spaniard, patting him on the head, said, with much candour, he thought his colour made no great difference.

Figure 8.9: Paragraphs shown to a user interested about history, with a compression rate of 33%. The uncompressed paragraphs are shown in Figure C.1.

History

After having been twice driven back by heavy south-western gales, Her Majesty's ship "Beagle," a ten-gun brig, under the command of Captain Fitz Roy, R.N., sailed from Devonport on the 27th of December, 1831. The object of the expedition was to complete the survey of Patagonia and Tierra del Fuego, commenced under Captain King in 1826 to 1830– to survey the shores of Chile, Peru, and of some islands in the Pacific– and to carry a chain of chronometrical measurements round the World. On the 6th of January we reached Teneriffe, but were prevented landing, by fears of our bringing the cholera: the next morning we saw the sun rise behind the rugged outline of the Grand Canary Island, and suddenly illumine the Peak of Teneriffe, whilst the lower parts were veiled in fleecy clouds.

One day, two of the officers and myself rode to Ribeira Grande, a village a few miles eastward of Porto Praya. Until we reached the valley of St. Martin, the country presented its usual dull brown appearance; but here, a very small rill of water produces a most refreshing margin of luxuriant vegetation. In the course of an hour we arrived at Ribeira Grande, and were surprised at the sight of a large ruined fort and cathedral. Having procured a black Padre for a guide, and a Spaniard who had served in the Peninsular war as an interpreter, we visited a collection of buildings, of which an ancient church formed the principal part.

We returned to the Venda to eat our dinners. We presented the black priest with a few shillings, and the Spaniard, patting him on the head, said, with much candour, he thought his colour made no great difference. We then returned, as fast as the ponies would go, to Porto Praya.

Figure 8.10: Paragraphs shown to a user interested about history, with a compression rate of 45%. The uncompressed paragraphs are shown in Figure C.1.

Technique	Description
Direct guidance	The arrows that let the user navigate to the previous and next section can be considered as the *next* links that provide a way for the user to read all the information without getting lost in the hyperspace.
Link hiding	When a link points to a section that does not contain any relevant information for a particular user, that link is hidden.
Link annotation	The colours in the background of the links that are found in the text tell the user which is the kind of information that they contain: whether they point to a date, a location or a person.
Link generation	Whenever the system has to write the name of a person, location or date for which there is information available that is relevant to the user, a hyperlink will be generated toward that information.

Table 8.8: Adaptive navigation support techniques that have been implemented for the user interface of the system.

chronologically. A small calendar will appear, from which it is possible to look for a specific date; there will a hyperlink for every date for which there is any information available. Whenever a date in the text (a yellow hyperlink) is chosen, this calendar is automatically updated to show the context of that date, and the text is updated to show all the information relative to that date. The calendar is displayed in Figure 8.11.

On the other hand, if the user is choosing the arrows for navigating a document in the order in which it was written as a linear text, then the left frame will show an enumeration of all the chapters and sections that were identified in the texts, and for which there is relevant information for the user. The sections that do not contain relevant data will not appear in this list, even without a hyperlink. An example of this index frame is shown in Figure 8.12.

In addition, for every paragraph in the text there are two small icons, that can be used by the users to inform the system that the information in that paragraph is relevant or irrelevant to them. This information will be used to update the frequency counts in the topic signatures in the users' profile, in order to provide better information in the future.

Finally, one link at the beginning of every paragraph that has been summarised allows the user to see the paragraph as it was originally. This is useful in the case that the summary has produced any inconsistency, such as omitting the antecedent of a pronoun that is used afterwards. If this happens, the user can substitute the summarised paragraph by the original version by clicking the link. A new link appears then that allows the user to restore the summarised paragraph.

The adaptation techniques that have been implemented are summarised in Tables 8.8 and 8.9. Please refer to Chapter 2 for a detailed description of these techniques.

8.4 Gathering Information from the Internet

As an additional functionality, the contents of the site can be extended with additional information, gathered from the Internet, which the users can optionally read, and which is collected from different sources other than

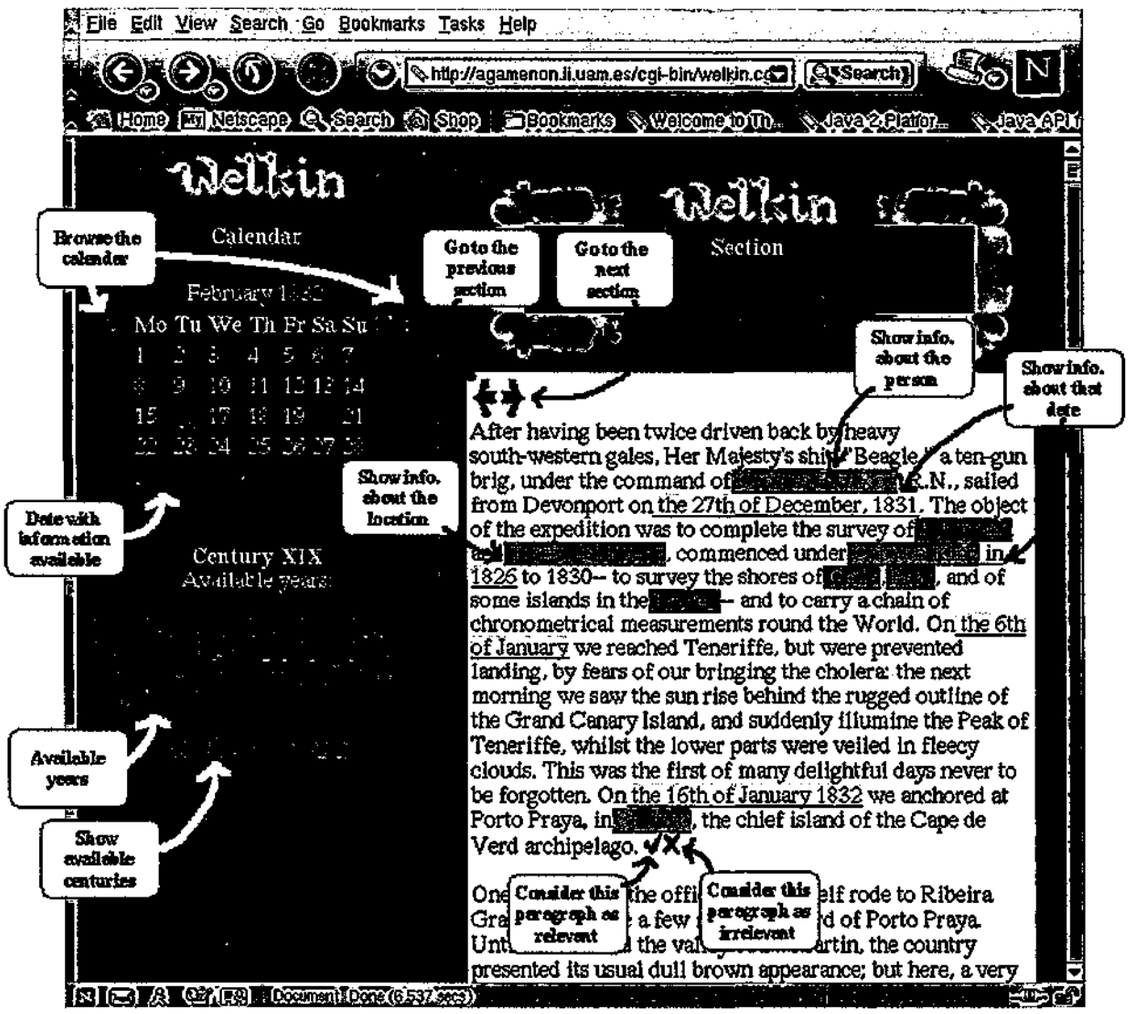


Figure 8.11: WELKIN main page showing a section of *The Voyages of the Beagle*.

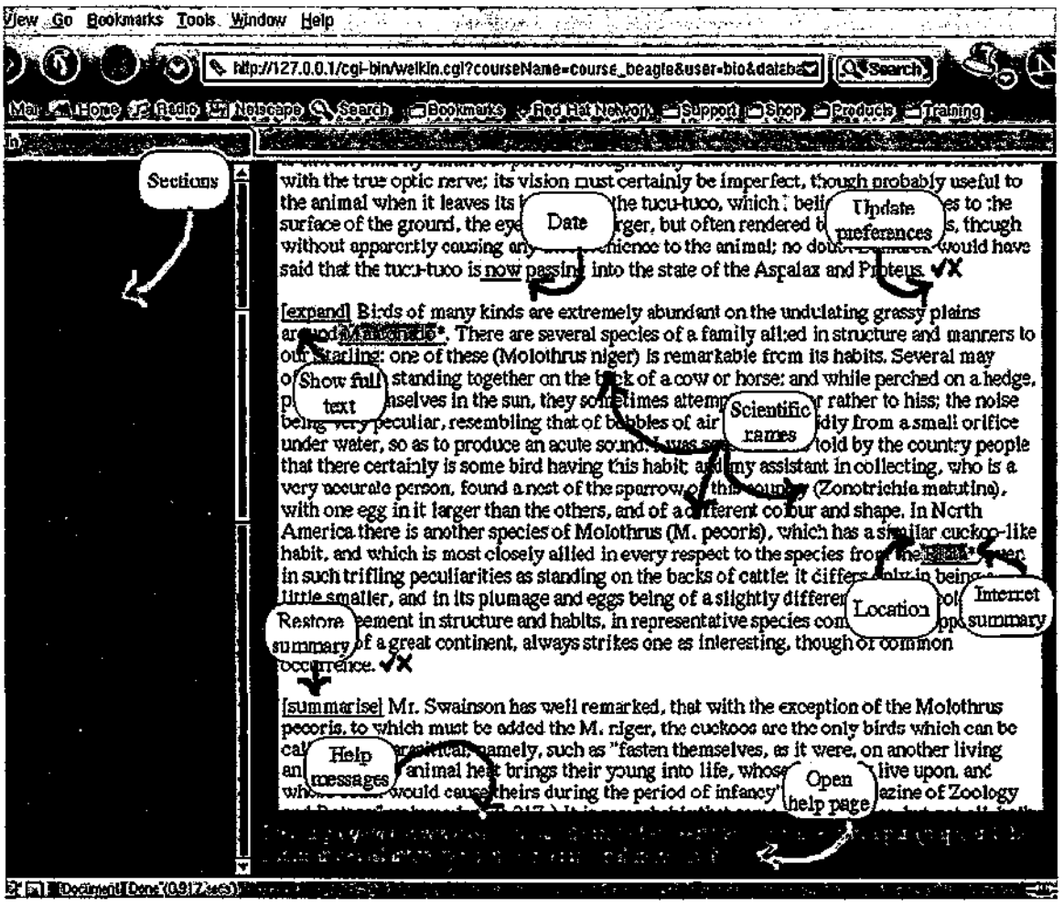


Figure 8.12: WELKIN main page showing a section of *The Voyages of the Beagle*.

Technique	Description
Removing fragments	All the paragraphs that contain information that is not relevant for the user are removed. If a section does not contain any relevant paragraph, it is removed as well, so the user does not even know that the section was there.
Altering fragments	The information in the relevant fragments is summarised by generating an extract of the sentences they contain, in a way that is tailored to the user's interests.
Stretchtext	Special links allow the user to expand or summarise each of the paragraphs that have been summarised.
Content annotation	Special colours are used in order to highlight relevant entities in the text, such as scientific names.

Table 8.9: Adaptive presentation techniques that have been implemented for the user interface of the system.

the original documents. For this aim, an additional module has been built that performs the appropriate searches on the Internet, and applies some filters as described below. A multi-document summarisation system that uses a few heuristics to find the relevant information can perform this step.

The summary pages will be generated for each one of the topics that were identified in the Term Identification and Classification step (see Chapter 5 for a description of these steps). This may include all the high-frequency proper nouns found, such as names of locations, people, rivers, etc; and the domain-specific common nouns that were identified, such as artifacts or animals.

As mentioned at the beginning of Chapter 7, summarisation systems can be classified taking into consideration several parameters. The following are the values of the parameters in the particular case mentioned here:

1. **Compression rate.** In this case, the compression rate is not fixed. Instead, it will depend on the quality of the pages found on the Internet. If most of that information is considered irrelevant for the purposes of the generated hypermedia site, then the generated summaries will be very short, because there will not be enough source information from which to construct the summary. On the other hand, if many pages with much important data are found, the summaries will be long.

2. **Audience.** The audience consists of the potential users of the system. Therefore, the summary will have to contain information that refers to the same topics found in the original documents from which the hypermedia site was constructed.

3. **Relation to source.** The generated summaries will consist of extracts of portions from the documents found, instead of a multi-document abstract. In general, abstracting is a more complicated task that requires an in-depth analysis of the text so as to be able to reformulate the sentences in a more condensed way. On the other hand, extracting can be performed with a shallow approach, which can be more robust and portable across domains, but which presents other disadvantages, such as the difficulty of keeping the coherence between the extracted fragments. Several heuristics are used in order to guarantee at least a level of coherence that makes the summaries readable.

4. **Coherence.** For the moment, we only intend that the extracted paragraphs, considered separately, are coherent units, although the reordering of the extracted paragraphs in the summary may not be completely coherent. That will be addressed in the future.

5. **Span.** As already said, the summary spans through several original documents (it is a multi-document summarisation system). The text source is a collection of roughly one hundred documents downloaded from Internet, from which a single summary page has to be obtained.

6. **Language.** The current implementation is monolingual, like the rest of the system, but none of the procedures used cannot be ported to other languages.

7. **Genre.** The system should be able to process the kinds of informations that are more frequently found in web pages. Narrative and descriptive text, together with tables and itemised lists can currently be processed, but special environments such as poetry or complex layouts with stylesheets have not been taken into account yet.

The approach taken for generating a summary page about some topic, such as a particular location, is shown in Figure 8.13. The next sections describe each part in detail.

1. Firstly, a search is performed with the Google (http://www.google.com) Internet search engine, using special keywords in order to increase the probability that the retrieved pages are relevant to the topic in question.
2. Secondly, the pages provided by the search engine, up to a maximum of one hundred documents, are retrieved from the Internet.
3. The format of the pages is changed into XML, and standard linguistic processing is performed on them, including tokenisation, sentence splitting, chunking and shallow parsing.
4. A multi-document coreference resolution is applied, but only to the topic of interest (e.g. the name of the particular domain-specific term). All the documents that contain that topic, but used with a different meaning, are filtered out.
5. Finally, some heuristics are used in order to extract the relevant information from each source, and all the paragraphs are put together in a single summary document.

Figure 8.13: Algorithm for collecting accurate additional information from the Internet.

8.4.1 Internet Search

In order to perform the search on Internet, a query is generated that contains the words of the topic of interest, e.g. *John Smith* and *Mr. Smith*. Additionally, some optional keywords are added to the query. These keywords are all the other domain-specific terms that appear in the same sections as the concept of interest, in order of frequency of appearance.

After performing different experiments, it was found that by adding all the domain-specific concepts that appear in the context, an undesirable consequence happens: many of the documents retrieved from the Internet are copies of the original documents from which the hypermedia site was created, and therefore they do not provide any new information to the user.

It was observed, on the other hand, that, if only the domain-specific concepts that had been classified as locations are added to the query, then the retrieved documents display a richer variability, and it is possible to find information about the changes that a particular person or location has experimented with the years. Therefore, in the final implementation, only the locations that appear in the same sections as the relevant term are included in the query.

For illustration, let us consider the concept *Valparaiso*, a city in the coast of Chile which appears in *The Voyages of the Beagle*. All the examples throughout this section will be described with this concept, so it is easier for the reader to follow them. *Valparaiso* had been automatically classified as a location. In order to generate a summary of this concept, the following query was sent to Google:

> "Valparaiso" ("Chile" OR "Chonos Archipelago" OR "S. Carlos" OR "Coquimbo" OR "Talcahuano" OR "Copiapo" OR "Concepcion" OR "Callao" OR "Cordillera" OR "Bahia Blanca" OR "Banda" OR "Peru" OR "Lima")

Using this query, one hundred documents were downloaded, translated into XML, and processed in order to obtain a shallow syntactic analysis of the sentences.

8.4.2 Relevant-term multi-document coreference resolution

After collecting the information, it is necessary to make sure that the documents are really relevant about the domain specific concept, in the context of the hypermedia site.

For example, it may be the case that *Valparaiso* is the name of a company, or it is included in a more complex name (e.g. *University of Valparaiso* or *Port Valparaiso Authorities*). It would be desirable to filter out all these other senses of the word before generating the summary. In our experiment, there were a couple of documents concerning a different city called *Valparaiso*, which is located in Indiana, in the U.S.

In order to discover which of the citations of the concept of interest are used with the requested meaning, contextual and co-occurrence information was also used: from the original documents, the system retrieves again all the paragraphs where the concept is used, and extracts the domain-specific terms that are located in them, and their frequencies of appearance. If it is helpful to do so, this vector may be considered as a *concept signature*, in relation to the topic, subject and object signatures introduced before in Chapter 4.

Next, the concepts in the vector were grouped using the *gloss* equivalence relationship $\mathcal{G}$: two concepts c and d are considered to be related in the following cases:

$$c \text{ appears in the definition of } d \Rightarrow c \,\mathcal{G}\, d \tag{8.11}$$

$$c \,\mathcal{G}\, d \Rightarrow d \,\mathcal{G}\, c \tag{8.12}$$

$$\exists e : c \,\mathcal{G}\, e \wedge e \,\mathcal{G}\, d \Rightarrow c \,\mathcal{G}\, d \tag{8.13}$$

The next step is performed with the WordNet ontology. The only relationship that shall be taken into account is the gloss relationship, as defined above. Therefore, the concepts from the signature can be grouped according to the connected subgraphs that they form; these subgraphs are the equivalence classes produced by the relationship $\mathcal{G}$.

For example, the following is the definition of *Chicago* in WordNet:

1. Chicago, Windy City – (largest city in Illinois; located on Lake Michigan)

Therefore, there is a gloss relationship between *Chicago*, *Illinois* and *Lake Michigan*. If only Chicago appears in the documents, with a frequency of 2, then all the other locations can be considered to have a frequency of appearance *zero*, and the weight of the subgraph can be calculated as the sum of the frequencies of all its members.

The following is the connected subgraph that was obtained for the concept *Valparaiso*, with the original documents:

{(Mexico,1)},
{(Buenos Ayres,1)},
{(Banda,1)},
{(Saint Lucia,1)},
{(Mendoza,1)},
{(Chonos Archipelago,1)},
{(Guayaquil,2)},
{(Copiapo,7)},
{(Aconcagua,3)},

{(Australia,1)},
{(Calabria,1)},
{(England,1)},
{(Talcahuano,3)},
{(S. Carlos,1)},
{(Madeira,1)},
{(Lisboa,1)},
{(Chimborazo,1)},
{(Callao,4)},
{(Chiloe,2),(Brasil,1),(Central America,1),(Tierra del Fuego,1),(Lima,3),(Peru,1),(America,1),
(Patagonia,3),(Valparaiso,29),(Concepcion,5),(Gran Santiago,3),(South America,3),
(Chile,10),(Pacific,1)},
{(Bahia,1)},
{(Coquimbo,5)}

As can be observed, there is one subgraph that contains many more concepts than the rest, and which also contains the concept whose summary we wanted to obtain:

{(Chiloe,2),(Brasil,1),(Central America,1),(Tierra del Fuego,1),(Lima,3),(Peru,1),(America,1),
(Patagonia,3),(Valparaiso,29),(Concepcion,5),(Gran Santiago,3),(South America,3),
(Chile,10),(Pacific,1)},

Secondly, for each document downloaded from the Internet, the same process is done: all the relevant concepts found in those documents are selected, and the subgraphs are generated in the same way. The following are the connected subgraphs in the locations graph for a document that refers to *Valparaiso* in the United States, instead than in Chile. The connected subgraph with the highest weight here is the one with refers to the United States (with weight 15); while the subgraph referring to Chile only has weight 1.

{(Columbia,1)},
{(Chicago,2),(Lake Michigan,2)},
{(Chile,1)},
{(Middle West,1),(Detroit,1),(Everglade State,1),(Mexico,1),
(South Bend,1),(Hoosier State,3),(United States,7)},
{(Cornhusker State,1)},
{(Brasil,1)}

The heuristic used consists in that a document is accepted only if the most weighty subgraphs have shared locations. Otherwise, it will be filtered out and its contents will not be considered for the summary. Therefore, the system compares the most weighty subgraphs obtained from the original documents to the most weighty subgraph from each retrieved web page. Only when their intersection is non-empty the document is accepted.

8.4.3 Generating the summary

A manual observation of the documents that were accepted after the previous filtering concluded that most of them contained information about the concept, but much of that information was not relevant for the

user of the hypermedia site, such as special offers of travel agencies; trip logs; etc.

However, it was observed that with a very simple heuristic it was possible to rule out practically all that information, without much loss of relevant information. This heuristic consists in selecting the sentences where the concept appears as subject position, and retain all the subsequent sentences up to the next end-of-paragraph. If the concept does not appear in the subject position, then the paragraph is ignored. This heuristic allowed us to rule out more than 75% of the downloaded pages that did not contain relevant information, such as travel logs to a particular location, or travel agency advertisements.

Figure 8.14 shows the summary page generated for the city *Valparaiso*, and Figure 8.15 shows the paragraphs that were discarded as they refer to a city in the U.S.

8.4.4 Evaluation

Summarisation algorithms are usually evaluated in terms of *recall*, the percentage of relevant information that was selected for the summary. Other metric that can also be interested in this case is *precision*, the percentage of the selected information that was also relevant. Our approach, which consists on successive filterings of the web pages and on the paragraphs, is intended to maximise precision, even though recall may suffer a little if relevant pages are filtered out.

As can be seen in the example of *Valparaiso*, all the times it is mentioned in the final summary are relevant, and the concept is being used with the required sense; there is no spurious information corresponding to other meanings of the word *Valparaiso*. On the other hand, the summaries generated for words that are specially polysemous did contain spurious information, even after all the filterings. For example, there are usually many different locations and people with names of saints, such as *St. Paul*, *Santiago*, *St. Jago* or *St. Fe*. This makes it very difficult to discover whether a web page is referring to the same place that was cited in the original documents, and the level of precision usually falls as some irrelevant pages sneak into the summary. This evaluation tries to capture the strong and the weak points of this approach.

Table 8.10 shows the results for five different relevant concepts extracted from *The Voyages of the Beagle*. The first five lines describe:

- The number of pages that were downloaded for each concept.
- The number of pages that contained some information worth to appear in the summary (evaluated manually).
- The number of pages from which information was actually extracted; between parenthesis, the number of them that should not have been selected (the number of errors).
- The number of paragraphs, from the relevant pages, that contained information worth to appear in the summary.
- The number of paragraphs from which information was actually extracted; again, the number between parenthesis is the number of mistakes.

With this information, it is possible to calculate the recall and precision concerning the choice of web pages and of paragraphs in them. These values appear in the last four lines in the table.

Table 8.11 describes some of the errors that provoked the drop in recall. Most of the errors were due to the following three causes:

Valparaiso

Valparaiso's hills and railways are not the only similarities it bears to California's San Francisco ; over the last 90 years, earthquakes have come and gone, one of them completely devastating the city in 1906–the same year as San Francisco's Great Quake.

Vina del Mar and Valparaiso are twin cities located on the Chilean coast. Valparaiso, the second-largest city and largest port city in Chile, was once a famous port-of-call for ships rounding Cape Horn in the 180ss. The elegance of this picturesque, hilly town - with twisting streets, Victorian houses, and funiculars that transport pedestrians up the slopes - is an artist's delight. The Chilean Congress has been moved here from Santiago, providing new momentum to this city.

Valparaiso is the principal port of Chile and with its resort companion to the north, Vina del Mar, offers an attractive destination for the cruise passenger. Santiago, the capital, is some 70 miles to the southeast within the foothills of the Andes.

However, Valparaiso's climate is generally mild, and thousands of tourists visit the region, particularly nearby Vina del Mar. Valparaiso was founded in 1536 by the Spanish conquistador Juan de Saavedra but was not permanently established until 1544 by Pedro de Valdivia. It was frequently raided by English and Dutch pirates throughout the 16th and 17th cent. Relatively unimportant in colonial times, the city grew in the late 19th cent. It has several museums, a Catholic university, a technical school, and a naval academy.

Valparaiso, the main port of Chile, was discovered in 1536 and has a long history. The city is the home of the National Congress and the Chilean Navy.

As the center of administrative service, Valparaiso has 45 hills and possesses a unique atmosphere with these hills, which make up 95 percent of the city. With its abundant natural resources, main trade items are copper, fruit, gas, petroleum and grains, and the local economy is mainly related to the port industry.

Figure 8.14: Generated page about the concept *Valparaiso*.

> Valparaiso has a variety of quality buildings, and greenfield sites ready for development. These sites are located along the 49 Bypass, both north and south of US 30. Some sites are already developed as industrial/commercial parks that are subdivided with all infrastructure in place. Some parks are located near the Porter County Airport for easy access to business air service.
>
> Valparaiso has developed a balanced business community that currently includes international companies that produce parts for the computer age, and a national company that makes Orville Redenbacher popcorn. These companies and many more have found the Hoosier work ethic exceptional, and the cost of doing business in Indiana modified by state incentives, and a frozen tax levy.

Figure 8.15: Paragraphs discarded about the concept *Valparaiso*, which refer to a different city.

Concept	Valparaiso	Cordillera	Patagonia	Buenos Ayres	Chiloe
Pages downloaded	100	89	95	91	98
Pages with information	18	8	11	2	13
Pages selected	13(0)	2(0)	5(0)	2(2)	8(0)
Relevant Paragraphs	29	14	11	2	34
Paragraphs Selected	16(0)	3(0)	5(0)	2(2)	13(0)
Page recall	72.22%	25%	45.45%	0%	61.54%
Page precision	100%	100%	100%	0%	100%
Paragraph recall	55.17%	21.43%	45.45%	0%	38.24%
Paragraph precision	100%	100%	100%	0%	100%

Table 8.10: Results for the multi-document summariser for information collected from the Internet. Between parenthesis are the number of selected pages and paragraphs that were incorrect.

Error type	**Number of times**
Concept not as subject	12
Syntax analysis error	11
Unknown synonym in the text	8
Incorrect processing of other languages	4
Indirect reference to the term	2
Pronoun coreference not solved	1
Page incorrectly filtered out	1

Table 8.11: Reasons for the recall errors committed by the algorithm.

- In twelve occasions, the relevant concept did not appear as subject, and the information contained in those paragraphs was rejected.
- In eleven occasions, the syntax analysis was incorrectly performed, and although the concept was in the subject position it was not recognised as such.
- Eight other pages referred to the relevant concept with a synonym that was not known by the system, so they were rejected. For example, in several occasions the *Cordillera* (the Spanish for *mountain ridge*, with which Darwin refers to the Andes mountains) was referred to as *Andes* in the web pages, which was a name not known.

A few number of errors happened because of other causes, such as the case when a portion of the web page contained words in a different language rather than English, when the concept was referred to with a pronoun, or when a page was incorrectly rejected using the procedure of the connected graphs. In two other times it happened that the relevant concept was not explicitly mentioned; instead, the author of the document referred to in an indirect way, and it would be necessary to perform an in-depth discourse analysis, including pragmatics interpretation, to be able to discover it.

8.5 Summary and discussion

In order to generate information that is adapted to the user needs, it is necessary to model the characteristics of the user that will influence the generation of the contents and the structure of the hypermedia site. In this approach, the following information has been taken into account:

- User interests.
- Compression rate for the texts.

A new approach has been implemented for automatically adapting a complete web site to *the interests of a user*. The users can specify their needs by choosing from a list of pre-defined interests, or by selecting, from a set of paragraphs, those which are of interest to them. Internally, the user preferences are represented with topic signatures, which include all the words that appear in some paragraphs classified as relevant for some field of knowledge. Users can always maintain their own profiles by classifying some information as relevant or irrelevant while browsing the information. For each hypermedia document, its contents will be filtered with a topic identification module that checks whether each paragraph is relevant to the users according to their signatures. Every paragraph judged irrelevant will be discarded from the output page; if a page is left empty, it is removed from the user's hyperspace until a change in the user's profile makes it relevant again.

Secondly, by indicating a compression rate to be performed to the pages, their contents are summarised using a sentence extraction procedure, which also takes into account the user's interests. The rate can be specified directly, or by means of the total number of words in the generated site or the user's availability of time. In this last case, it is necessary to know the user's **reading efficiency**, which is measured with a web page that performs a *reading speed* and a *reading comprehension* test.

This technique performs well for web sites that contain large amounts of text, such as those that have been generated from linear text. In this case, the system selects just the paragraphs that meet the user's interests. All this information can be used to determine which compression rate should be performed to the hypermedia site contents so the user can browse the whole contents of the site in the time available.

A new summarisation algorithm has been described, based on generating extracts of texts with genetic algorithms. The method is very easy to program, and the performance is good considering its complexity, as it takes in average 250 milliseconds to summarise a 20-sentence text in a Pentium III 900MHz machine. In contrast, Marcu [2001] states that an algorithm with similar weight functions takes a couple of hours of computation for each document collection in the Document Understanding Conference. A more detailed discussion on the algorithm can be found in Section 10.4.4.

Finally, the information can be extended with additional data about the relevant concepts in the site, taken from the Internet. This step can be performed off-line, after creating the database of the generated site. A set of one hundred documents are downloaded for each concept, and several steps of filtering are performed in order to eliminate the irrelevant data. The paragraphs that are considered relevant by all the filters will be shown to the user in a single web page.

Part IV

Evaluation and Conclusions

Introduction to Part IV

After the description of the system, the following chapters describe the evaluation and end with a description of the contributions of the thesis and the lines open for future work.

Concerning the evaluation, two different experiments have been done. The first one concerned assessing the performance of the system for creating hypermedia sites from linear texts. Three different texts have been chosen for building example sites: Darwin's *The Voyages of the Beagle*, Osler's *The evolution of modern medicine*, and Hegel's *Lectures on the history of philosophy*. The second experiment was performed with users in order to collect their opinions about the system. The descriptions and results of the evaluations can be found in Chapter 9.

Finally, Chapter 10 contains the final conclusions and some ideas about how these techniques can be improved.

Chapter 9

Three case studies and a usage evaluation

For evaluation purposes, three different hypermedia sites have been built, from three texts: Darwin's *The Voyages of the Beagle*, Osler's *The evolution of modern medicine*, and Hegel's *Lectures on the history of philosophy*. There are several reasons for the choice of these texts. Firstly, they deal mostly with historical events, which can be studied from different points of view and with different purposes, a requisite so that different stereotypes can be defined for them. Secondly, they are smoothly written, with few syntactic errors, and very few orthographic errors, which makes them apt for being processed with the linguistic tools. Finally, they contain mostly physical entities, for which the system has collected contextual signatures, so the Term Classification module can be applied to them as it is. The generation of the three hypermedia sites is described in Sections 9.1, 9.2 and 9.3.

At the end of the chapter, Section 9.4 describes an evaluation that was performed with users in a controlled experiment, which included two tasks for which they needed to use the sites about Darwin's and Hegel's texts, respectively. The section includes some reflections about the result.

All the experiments have been performed using a Pentium III 900 MHz PC, with 256 Mbytes of memory. This includes both the off-line generation of the databases for the adaptive site, and the server that was used remotely by the users to try the system in the usage experiment.

9.1 Charles Darwin's *A Naturalist's Voyage round the World*

In December 1831, Charles Darwin embarked the *Beagle* for a world-round trip. The voyage had been unexpected, as he had only had a few months for the arrangements. The previous August, he had not even heard about the *Beagle* or the Captain FitzRoy, who commanded the ship. The trip had several purposes, such as improving the cartography of South America and improving the ways to determine the longitude while on board; but one of the inducements for FitzRoy to undertake the voyage was to return to Tierra del Fuego several natives that he had taken to England in a previous trip, and he had educated at his own expense. Unexpectedly for Darwin, he was offered the post of naturalist in the ship, and he decided to join the expedition. During the voyage, Darwin had many occasions to study animals and plants, specially around South America and the Galapagos Isles, and next in New Zealand and Australia, before going back

to the British Isles in 1836, sailing around the Cape of Good Hope. The travel provided him with material for several years of study, product of which was his Theory of Evolution as Natural Selection.

A few years after returning to England, Darwin published *A Naturalist's Voyage around the World*, written as a travel log of the expedition. The book is a multi-disciplinary text in which Darwin described the voyage, including descriptions of the places visited, the animals and plants found therein, the folklore of the natives of the different countries and regions, and their own adventures. Therefore, it is not only a book appealing to biology students and amateurs; it also contains information regarding natural and political geography, and firsthand accounts of historical events. Hence, this book has been taken as a good material for building an information system that adapts to the preferences of the users.

The following sections describe the output produced by the different components of the WELKIN system:

9.1.1 Classification of new terms

The Term Extraction procedure found 89 unknown concepts and proper names that appeared in the source texts with a frequency of 50 or higher. These includes names of the people, locations, artifacts, rivers, mountain ridges, animals, etc.

After their identification, the system automatically collected the topic, subject and object signatures for each of them, and proceeded to classify them in WordNet. Most of the terms were correctly classified in the main groups (locations, bodies of water, people or animals), but there were some that weren't. The following are the most important errors in the classification that were produced:

- Some of the unknown terms, specially those that referred to islands, mountain ridges, and buildings, could not be correctly classified because they are classified in WordNet below the synset *object*, and no signatures had been collected for that sub-tree. As explained in Section 5.6, because of time and space limitations, information has been collected only for roughly 3,000 WordNet synsets, which include a representative sample of all the main groups of entities: locations, animals, people, artifacts and bodies of water. But they had not been collected for any of the entities under *object* and therefore no object was properly classified.

  In theory, there is no reason why these objects should not be placed in their correct position if we had the resources available to collect all the remaining signatures. In fact, islands were all classified as bodies of water, e.g. seas and lakes, because most of the context words that appear in reference to the islands refer to the sea, a result which does make sense.

- Again as expected, it was difficult for the algorithm to make decisions when it arrived to parts of the WordNet ontology where the semantic distinctions between synsets are subtle. A particular case was when deciding the male/female distinction for persons. In this case, hyponymy patterns usually help, but sometimes none of these patterns were found. Table 9.1 shows the classifications gotten for several high-frequency people names that were selected from the text.

  It might be argued that this distinction can be easily done with a couple of additional resources. A list containing personal abbreviations such as *Mr.* and *Mrs.*, with gender information, might determine that some of these terms are male and others are female. Secondly, a gazetteer with first names can also help to distinguish male and female people. Finally, WordNet might be extended with information about the gender of the synsets. These modifications could be easily added to the algorithm, and then more than half of the errors might be corrected. It has not been done because, in that case,

person	Classified as	Sex correct
Captain FitzRoy	**shaver**: an adult male who shaves	yes
Azara	**boy**: a friendly informal reference to a grown man	yes
Humboldt	**married woman**	no
Captain Sullivan	**boy**	yes
Mr. Bushby	**boy**	yes
Mr. Lyell	**married woman**	no
Mr. Waterhouse	**Methuselah**: a man who is very old	yes
Captain King	**boy**	yes
Jemmy Button	**married woman**	no
Captain Cook	**boy**	yes
Dr. Richardson	**applicant**: a person who requests or seeks something such as assistance or employment or admission.	yes
York Minster	**boy**	yes
Mr. Low	**boy**	yes
Mr. Beanow	**sweater girl**: a girl with an attractive bust who wears tight sweaters.	no
Matthews	**mayoress**: the wife of a mayor	no

Table 9.1: People that appear in the text, and the way in which they were classified.

the algorithm would lose the property that it is fully unsupervised and can be used, with no human supervision, for any ontology.

9.1.2 Time expressions detection

The time expressions specialist found 253 dates in the text, and 446 relative time expressions (such as *now*, *today*, *yesterday* or *three years ago*). In total, there were 699 temporal expressions recognised.

Out of them, 558 expressions could be analysed and resolved. Some of the remaining relative expressions (e.g. *the next morning*) could not be resolved because there was not enough contextual evidence. Some complex temporal expressions, such as *from 1850 to 1860*, went unrecognised because they did not match with any of the regular expressions used.

9.1.3 Scientific names identification

The algorithm for scientific names identification was applied on *A Naturalist's Voyage round the World*. After collecting all pairs of words such that the first one is capitalised and the second is not, all the pairs that contained any common English word were ruled out. That left a total of 102 pairs of words from the whole book. Of these, after using the Latin-endings and the language identification filters,

- 83 were correctly accepted (true positives).
- 3 were incorrectly accepted (false positives).
- 13 were correctly rejected (true negatives).
- 5 were incorrectly rejected (false negatives).

Task	Time
Format change into XML	<1 minute
Tokenisation	<1 minute
Sentence splitting	<1 minute
Part-of-speech tagging	1.5 minutes
Stemming	<1 minute
Time expressions recognition	<1 minute
Scientific names recognition	62 minutes
Cascade of three chunkers	20 minutes
Quotes solver	1 minute
Parser	1 minute
Term Identification and Classification	56 minutes
Document structure analysis	4 minutes
Time expressions resolution	<1 minute
Course database creation	5 minutes
TOTAL	~2.5 hours

Table 9.2: Performance of each of the modules that are executed in order to create automatically a hypermedia web site from a linear document.

Overall, these results show that the accuracy of the algorithm is 94.12%, and the recall is 94.32%.

Furthermore, it found 39 additional appearances of known genuses in the text, without the species name: these were marked as well. There were, in the texts, some names of genuses that do not appear together with a species name. Due to the characteristics of the algorithm, all these could not be identified. The text contained many unknown capitalised words, only some of which were really genus names, and a single word is usually too short for the language identification module to be reliable. Therefore it would probably be necessary to use a Machine Learning algorithm that take into consideration the context of these genus names in order to identify them.

The three false positives were due to the following cases:

- A pair of Spanish words embedded in the text, *Observa sobre*, was classified as Latin.
- Two pairs of Latin words that were not part of a scientific name, but part of a quotation, *Per somnum* and *Deum laudamus*.

Note that all the analysis that the system is doing is morphological: unknown words that have certain endings or certain N-grams. Morphologically, any of these false positives could be scientific names; it would be necessary to perform a context analysis to check that they are part of a larger sentence in a foreign language in order to rule them out.

The five false negatives were true scientific names that were classified as English because they contained several times the sequence *th*, which happens to provide more support to the classification as English.

As a final remark, it must be born in mind that the distinction between Latin and English is probably easier than between Latin and a romance language, such as Spanish, so it is not clear, without empirical evidence, whether this algorithm will be portable across languages.

9.1.4 Performance of the off-line processing

All the off-line processing was performed in roughly two hours and a half in the above-mentioned computer. Table 9.2 shows the amount of time that was required by each of the subtasks performed. As can be

seen, only three modules spend together most of the processing time. The scientific names recogniser is the slowest one; the reason of the long time that it takes is that it has to execute a system call to the language identifier for each pair of words such as one is capitalised and the other is not; if the language identification module were implemented in Java, then the system call would not be necessary, and it could be greatly sped up.

The other two modules that need much time to execute are the cascade of chunkers (that identify Quantifier Phrases, base Noun Phrases and Verb Phrases), which apply a long list of transformation rules to every sentence in the text; and the Term Identification and Classification module, that involves the gathering of contextual data, and complex calculations.

9.1.5 Topic classification

When users connect to the system, they specify their interests in two possible ways: by choosing one of the suggested stereotypes, or by marking which paragraphs, from a pre-defined set, contain relevant information for them.

In the case of *A Naturalist's Voyage round the World*, as described in Section 8.1.1, three initial stereotypes were chosen:

- **Biology**, for users interested in animals and plants.
- **Geography**, for users interested in descriptions.
- **History**, for the users interested in the narrative portions of the text.

The first one hundred paragraphs in the book were manually classified in one of the three classes (biology, history and geography), and the algorithm was trained on that data. The whole process did not require more than one hour of work.

For illustration, Figures 9.1, 9.2 and 9.3 show the tables of contents of the first six chapters of the book that were generated for the three possible stereotypes. Note that, when a section does not contain relevant information for some particular stereotype, it is removed from the index and utterly ignored in the adaptive web site.

9.1.6 Summaries from Internet

Summaries were automatically collected for the 89 terms that had been identified and classified into WordNet, and for 68 additional WordNet synsets, that represent instances (e.g. locations or people), and which also appeared in the documents, such as *Spain*, *Scotland* or *Germany*. Of the 157 summaries generated in total, 37 were left empty, because no relevant information was found for them in the Internet pages. Either there was no page retrieved, all of them were filtered, or the term did not appear in subject position.

The mean time to collect a summary was 20 minutes in the above-mentioned platform, of which 30 seconds were necessary to study the term (study its context in the original documents and create the connected subgraphs using *gloss* links); eighteen minutes in average were needed to download the one hundred documents from the Internet and process them with the linguistic tools; and one minute and a half was needed in order to analyse the texts and produce the summary file.

In total, 52 hours were necessary to generate all the summaries, and therefore this step is the more time-consuming of all the processing involved in automatically generating a hypermedia site with WELKIN. It is

CHAPTER I.
1. ST. PAUL 'S ROCKS
2. BAHIA , OR SAN SALVADOR BRAZIL , FEBRUARY 29 , 1832
3. MARCH 18 , 1832
CHAPTER II
1. APRIL 9 , 1832
CHAPTER III
1. JULY 26 , 1832
CHAPTER IV
1. AUGUST 11 , 1833
CHAPTER V.
CHAPTER VI
1. SEPTEMBER 8 , 1833
2. SEPTEMBER 12 AND 13 , 1833
3. SEPTEMBER 15 , 1833
4. SEPTEMBER 17 , 1833
5. SEPTEMBER 19 , 1833

Figure 9.1: Table of contents of *A Naturalist's Voyage round the World* for a user interested in biology (first six chapters).

CHAPTER I.
1. ST. PAUL 'S ROCKS
2. FERNANDO NORONHA , FEBRUARY 20 , 1832
3. BAHIA , OR SAN SALVADOR BRAZIL , FEBRUARY 29 , 1832
CHAPTER II
1. APRIL 14 , 1832
2. APRIL 18 , 1832
CHAPTER III
1. JULY 5 , 1832
2. JULY 26 , 1832
CHAPTER IV
1. AUGUST 11 , 1833
CHAPTER V.
CHAPTER VI
1. SEPTEMBER 8 , 1833
2. SEPTEMBER 11 , 1833
3. SEPTEMBER 17 , 1833
4. SEPTEMBER 18 , 1833
5. SEPTEMBER 19 , 1833
6. SEPTEMBER 20 , 1833

Figure 9.2: Table of contents of *A Naturalist's Voyage round the World* for a user interested in geography (first six chapters).

CHAPTER I.
CHAPTER II
1. APRIL 4 TO JULY 5 , 1832
2. APRIL 8 , 1832
3. APRIL 9 , 1832
4. APRIL 13 , 1832
5. APRIL 14 , 1832
CHAPTER III
1. JULY 5 , 1832
2. JULY 26 , 1832
CHAPTER IV
1. AUGUST 11 , 1833
CHAPTER V.
CHAPTER VI
1. SEPTEMBER 8 , 1833
2. SEPTEMBER 10 , 1833
3. SEPTEMBER 11 , 1833
4. SEPTEMBER 12 AND 13 , 1833
5. SEPTEMBER 14 , 1833
6. SEPTEMBER 15 , 1833
7. SEPTEMBER 16 , 1833
8. SEPTEMBER 17 , 1833
9. SEPTEMBER 18 , 1833
10. SEPTEMBER 19 , 1833
11. SEPTEMBER 20 , 1833

Figure 9.3: Table of contents of *A Naturalist's Voyage round the World* for a user interested in history (first six chapters).

also the more space-consuming step, as all the documents that were downloaded and then annotated with linguistic information occupied nearly three gigabytes. Of course, most of the generated files are intermediate steps during the total processing and can be deleted when they have been used.

It was observed that the contextual clues taken from the source document were really useful for filtering the downloaded pages. This was specially important about concepts that can have multiple meanings. For example, the unknown term *Cordillera* is the Spanish for *mountain ridge*, and Darwin uses it to refer to the Andean ridge. The contextual keywords for Google, *Peru*, *Chile*, *Patagonia* and *Tierra del Fuego* were very successful for downloading mainly pages about the Andean Cordillera, instead than any other ridge in the world. Finally, the filtering that is done with the sub-graphs ruled out one page about a Catalan wine called *Cordillera*, produced by the "casa Torres". A church called "Cordillera Church", in Chile, and a Chilean province called "Cordillera" were also discarded by the filters.

It is worth to note that documents that referred to the correct term, but in a very specific meaning, were also filtered out, because of the heuristics used. For instance, a document about the Andean Cordillera in Bolivia gave as the weightiest connected subgraph

{(Ancohuma,1),(Illimani,1),(Illampu,1),(La Paz,2),(Sajama,1),(Bolivia,10)}

The document was discarded because it did not include any of the concepts that appeared in the context of Cordillera in Darwin's book, which include Chile and Peru –he did not go to the Andean territory that is now in Bolivia, so that portion of the cordillera is ignored in the summary.

The heuristics were in general good for avoiding travel accounts, travel agency offers, hotels, and other spurious information. However, an analysis of the results helped in identifying some of the weak points of this approach:

- There is no control whether the contents gotten from the Internet are exactly the same as the source documents used. In this particular case, *A Naturalist's Voyage around the World* is available on several sites in Internet, and therefore it is likely that it will be returned by the search engine, given that it contains the domain-specific term together with the context words.

  If the unknown term is widely used in many contexts, such as the name of a city, then there is a higher probably that the pages downloaded are different from the original text. This happened in most of the cases. However, when the term looked for is only referred in Internet in relation to the *Beagle*, as it is the case of *Captain Fitz Roy* or the bird called *Carrancha*, then the Internet summary will simply repeat information that was already available on the generated hypermedia site.

- Secondly, there is no control yet whether two different Internet sources provide the same information, so the same paragraph may appear twice or more times in the summary page.

- Thirdly, if an unknown term is very polysemous, as it is the case of same location names that can refer to tens of different cities, provinces, streets, etc., then the filtering performed is not accurate enough. That was the case of *St. Helena* or *La Plata*, for which most of the information collected referred to different places than the one Darwin visited.

- Finally, when a domain-specific term has a different meaning that is commonly used, it was also difficult for the system to distinguish them. This is the case of the Fuegian *York Minster*, that started the trip from England with Captain FitzRoy. Every page downloaded from the Internet referred to the cathedral of York, and some of that "irrelevant" information found its way into the summary page. This particular case might be avoided if a standard Named Entity recognition module was used, because

the usual meaning refers to a building, and the specific meaning refers to a person, so the two can, in theory, be distinguished with automatic methods.

9.2 William Osler's *The evolution of modern medicine*

Sir William Osler is probably the best known physician in the English-speaking world at the turn of the XX century, sometimes called the "most influential physician in history". Born in 1849 at Bond Head, Ontario, he graduated in Medicine in 1872, and soon started teaching in the university, where he became an expert in diagnosis of diseases of the heart, lungs and blood. His textbook called "The Principles and Practice of Medicine" (first edition in 1892) was considered authoritative for more than 30 years. Some of his greatest achievements are the following:

- He emphasised the importance of the state of mind of the patients in order to heal, and has been called the father of psychosomatic medicine.
- He took an important role in creating the system for postgraduate training for physicians that is still followed nowadays. In fact, he emphasised the importance that medical students spent time with the patients.
- His contributions to many fields in medicine are many and varied. Osler was the first to describe the morphology of platelets. Several illnesses and at least one parasite have received his name.

The text that was chosen for generating the hypermedia site, *The evolution of modern medicine*, is the manuscript of Sir William Osler's lectures on the topic, which were delivered at Yale University in April, 1913. It was going to be published when the war started and postponed the project, and the manuscript finally remained unfinished at Osler's death. Notwithstanding Osler's will that any unfinished work should be left unpublished, Harvey Cushing, Archibald Malloch and others finally prepared it for press. It was reprinted in 1963 and 1972, and it is being sold still nowadays in hardcover and as e-book. It is recognised as one of the best books for starting the study of medicine.

9.2.1 Classification of new terms

The domain-specific terms from Osler's lectures have less contextual clues for classification than the terms from Darwin's text. There are two reasons for this. Osler's book is smaller than Darwin's, and the domain-specific terms change drastically as the narration advances to different ages and civilisations. Most people and locations mentioned in a chapter are not likely to appear afterwards in any other chapter of the book. Therefore, this experiment shows the weak points of the classification algorithm when there are very few frequencies collected for the signatures of these unknown terms.

The experiments described in Sections 5.6 and 6.2.3 were done with terms that appeared at least 50 times in the documents; however, in Osler's text none of the concepts appeared so often. Figure 9.4 shows the results for classifying the concepts, grouped by frequency of appearance. As can be seen, the classification is more accurate if there is more contextual evidence about a concept. By examining the results of the classification, it is clear that the accuracy of the classification degrades as the number of appearances of the concepts gets lower. Amongst the concepts with a frequency higher or equal to 20, four out of twelve were misclassified. The seven top-frequency concepts were all correctly classified, and the errors are together

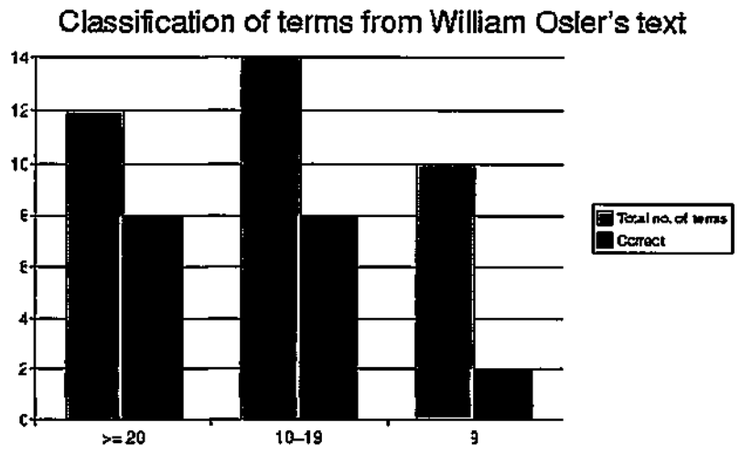


Figure 9.4: Results classifying new terms from Osler's *The evolution of modern medicine.*

between the concepts with lower frequencies. Below twenty, the number of errors increases: for those with a frequency of appearance between 10 and 19, six out of fourteen were misclassified; and those with a frequency of nine were mostly misclassified. The concepts with a frequency of appearance below nine were not even studied.

- An examination of the misclassified concepts, one by one, showed that, indeed, one of the reasons for these misclassifications is the fact that the signatures are not representative enough. As a few examples,
  - *Rome* only appears once as subject of a verb, and therefore there is only one item in its subject signature, which is the verb *to take* (with frequency 1). Of course, it is more probably to see a life form than to see a location as subject of that verb, and therefore it was classified as a life being.
  - *Morgagni* was correctly classified as a life being; but it was next classified as an animal, instead of as a person. By examining its signatures, it was found that it only appears once as the subject of a verb, and that verb is *to have*. Because a person can appear as subject of thousands of verbs; while animals have a fewer variety, the solitary appearance of the verb *to have* gives much more support to the choice of *animal* than to the choice of *person*, and that is why the misclassification occurred.
  - The remaining misclassifications can be explained with similar reasonings. A total of four people were classified as locations; three other people as animals, and one person was classified as a body of water. In contrast, thirteen people were correctly classified.

9.2.2 Performance results

Table 9.3 shows the performance results on Osler's *The Evolution of Modern Medicine.* As can be seen, the time required was very similar to the previous text, but with two differences:

- The scientific names recogniser was omitted from the processing, because it was not judged necessary for this kind of text.

Task	Time
Format change into XML	<1 minute
Tokenisation	<1 minute
Sentence splitting	<1 minute
Part-of-speech tagging	<1 minutes
Stemming	<1 minute
Time expressions recognition	<1 minute
Cascade of three chunkers	2.5 minutes
Quotes solver	<1 minute
Parser	<1 minute
Term Identification and Classification	39 minutes
Document structure analysis	11 minutes
Time expressions resolution	<1 minute
Course database creation	3 minutes
TOTAL	58 minutes

Table 9.3: Performance of each of the modules that are executed in order to create automatically a hypermedia web site from Osler's *The Evolution of Modern Medicine.*

Corpus	Distinct words
Voyages of the Beagle	2746
Evolution of Medicine	1892
Total	4637
Corpora together	3806
Words in common	831 (21.86%)

Table 9.4: Words in the topic signatures, taken from the first 100 paragraphs in Darwin's *The Voyages of the Beagle*, and words taken from the first 38 paragraphs from Osler's *The Evolution of Modern Medicine.*

- Osler's text was less than half in length, so the processing was done faster.

9.2.3 Topics classification

As could be expected, the stereotypes defined for the previous text are not the best choice for structuring *The Evolution of Modern Medicine*, because Osler's text does not deal with the same topics than Darwin's. Nevertheless, a first experiment was performed in order to discover which would be the results using the same stereotypes.

In this experiments, the first 32 paragraphs from Osler's text were taken. Those referring to anatomy or the biological aspects of pathologies were labelled as *biology*; the lives of researchers as *narrative*; and the descriptions of cultures and people as *geography*; and next it was used as a test set for evaluating the topic filtering. The accuracy of the automatic classification was only 43.75%; and the > 33.33% accuracy was 56.25%. A comparison of the words that appeared in the topic signatures, collected from Darwin's text, and the words that were in Osler's test set showed that only roughly 22% percent of the words were in common between both. This small lexical overlapping between the training set (extracted from Darwin's text) and the test set (taken from Osler's text) is probably the reason why the classification proved invalid. Table 9.4 shows the total number of distinct words in the training set and in the test set, and the percentage of overlapping between both sets.

For creating the corpus-specific stereotypes, all the hyponyms of the concept *medical science* in WordNet were considered as possible labels for classifying textual segments. There is a total of 80 subdivisions of the medical sciences in WordNet; however, not all of them were considered, because many of the medical specialities only appeared as the main subject a few paragraphs, or didn't appear at all, as was the case of *endodontics* or *nuclear medicine*.

Furthermore, while describing the state of medicine in the ancient civilisations, Osler added some explanations about the social organisations of ancient and different cultures, a topic that might be classified as *anthropology*. That is the reason why, although it is not a field inside medicine, anthropology was also chosen as one of the stereotypes.

In conclusion, the following four stereotypes were used for labelling 100 paragraphs from the text:

- **Anthropology**: The study of social organisation, myths, religion, codes of law and customs and uses of human cultures.
- **Anatomy**: The study of the parts and structure of the human body, or the body of animals.
- **Pathology**: The study of causes, nature and effects of diseases. It includes branches such as epidemiology, virology, etc.
- **General medicine**: The study of diagnosis and treatment of diseases. This label also includes all those branches of medicine whose areas of interest appeared scantly in the text, such as psychopathology, pharmacology, internal medicine or traumatology.

An evaluation with a test set produced results similar to the ones that had been obtained in Section 8.1.1. In this case, there are four stereotypes; therefore, when a user accesses the system, it calculates the similarity between each paragraph and the stereotype selected by the user. If that similarity is higher or equal to $\frac{1}{4} = 25\%$, then the paragraph is selected. The experiments produced an accuracy of 78.95%, and a $> 25\%$ accuracy (when the correct classification had a similarity higher or equal to 25%) of roughly 94%.

The manual annotation of 100 paragraphs in each of these four stereotypes, needed for training the system, took one hour and a half, a time that should be added to the total time needed for generating the hypermedia site.

9.3 Georg Hegel's *History of philosophy*

Georg Wilhelm Friedrich Hegel (1770-1831) belongs, together with Johann Gottlieb Fichte and Frederick Wilhelm Joseph Schelling, to the period of "German idealism" in the decades following Kant. Hegel lived in what is now Germany, although it was not reunited in a single state at the time. He taught in Tubingen, Jena, and Heidelberg, and finally ended, in 1815, at the University of Berlin, where he worked till the end of his days. He was a formidable critic of his predecessor Immanuel Kant. Hegel's theories about the interpretation of history have greatly influenced philosophers such as Marx.

The lectures on the History of Philosophy are based on Hegel's nine tenures lecturing on the history of philosophy: 1805-1806 in Jena, 1816-1818 in Heidelberg, and 1819-1830 in Berlin. They were first published between 1833-36 in volumes 13-15 in the first edition of Hegel's *Werke*. Initially published in English between 1892 and 1896, it is still sold nowadays.

The text that was processed for the hypermedia site is not the complete lectures, because it was not found on-line anywhere on the Internet. Only a portion of the third part of the book, the one concerning modern philosophy, was found and processed[1].

9.3.1 Classification of new terms

As in the case of Osler's lectures, a book about history is likely to change the topic as it advances through different periods of history, by naming different people and locations, and therefore most of the unknown terms appeared with a small frequency, so there was not much contextual information to guide the classification. Again, a total of 57 unknown terms or proper nouns that appeared with a frequency of 9 or more times in the documents were extracted and automatically classified. Below 9, the accuracy of the classification is very low.

The following are some observations that we may draw from the results of the classification:

- Eight of the concepts chosen by the Term Identification module were erroneous; some of them are abbreviations that appear frequently in the bibliography of the text, and others are abbreviations used to refer the reader to other sections.
- There were 22 concepts that referred either to books, or to abstract concepts, many of them multi-word German expressions. The system has not been trained for these kinds of concepts (signatures have only been collected for physical concepts), and therefore it was not able to classify them correctly inside WordNet.
- Most of the people and locations with a frequency of appearance higher or equal to 15 were correctly classified. The higher the frequency of the concept, the higher the confidence that the classification chosen is correct.

9.3.2 Performance results

Table 9.5 shows the performance results on Hegel's lectures. The text is shorter than *The Voyages of the Beagle* and, again, the step of scientific names recognition was not necessary, so the total processing time is less.

9.3.3 Topics classification

Two different stereotypes have been defined for Hegel's lectures: **philosophy** and **biography**. For every philosopher whose work is commented, Hegel includes a brief paragraph with his biography, and it was thought that it might be the only relevant information for some users.

A total of 54 paragraphs were annotated, from which 49 were labelled as dealing with philosophy, and 5 of them contained biographies. After applying the generated signatures with the whole text, a total of 15 biographies were found, all of which were correct.

In this case, one kind of stereotype users (the ones that chose biography as their interests) will only be shown a very small part of the whole web site; while the ones that select philosophy will be able to see nearly everything in the book, as philosophy is the main subject of the lectures.

[1] It was mirrored from `http://www.class.uidaho.edu/mickelsen/ToC/Hegel-Hist of Phil.htm`

Task	Time
Format change into XML	<1 minute
Tokenisation	<1 minute
Sentence splitting	<1 minute
Part-of-speech tagging	1 minutes
Stemming	<1 minute
Time expressions recognition	<1 minute
Cascade of three chunkers	7 minutes
Quotes solver	<1 minute
Parser	1 minute
Term Identification and Classification	37 minutes
Document structure analysis	3 minutes
Time expressions resolution	<1 minute
Course database creation	5 minutes
TOTAL	57 minutes

Table 9.5: Performance of each of the modules that are executed in order to create automatically a hypermedia web site from Hegel's *Lectures on the History of Philosophy.*

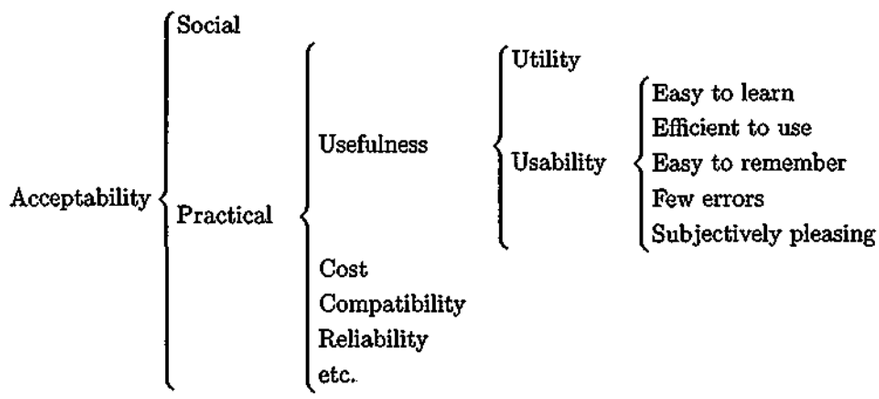


Figure 9.5: System Acceptability Attributes (Nielsen [1990], from Fritz [1995]).

9.4 Usage evaluation

The main objective of this research is to present some texts to users in a way that meets their interests; by condensing the texts and showing only the information that has been judged relevant. The aim of this work is the reduction of the time needed to read (or to learn) some information by removing the irrelevant fragments.

Adaptive Hypermedia can be considered a sub-area of Human Computer Interaction (HCI). HCI is a field concerned in the methods for building useful and usable interfaces between the computer and the user. Evaluation is a very important issue, as it must be tested that a particular adaptive interfaces is considered acceptable by the users. There are two main features that can be evaluated: **usability**, or the easiness of use of the system, and **utility** (sometimes referred as functionality), which tests that the system performs a useful task that meets some need of the users.

Figure 9.5 shows Nielsen's taxonomy of acceptability. Note that *usefulness* is divided into the two above-mentioned features: *utility*, meaning the adequacy of the system to the user needs, and *usability*, meaning

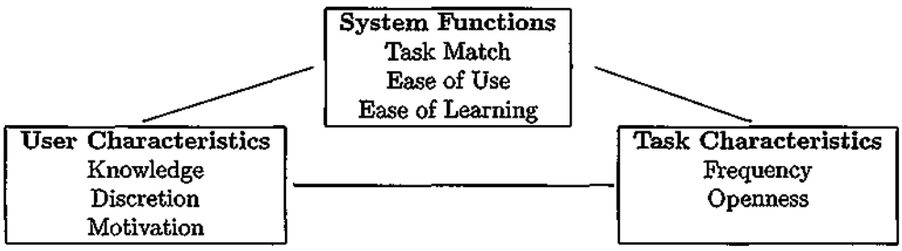


Figure 9.6: Eason's framework of usability (Eason [1993], from Fritz [1995]).

the easiness of use and appealness to the user. Nielsen further divides usability into several variables that can be measured empirically, such as easiness to learn and remember, efficiency of use, accuracy (few errors) and attitude (subjectively pleasing), but no variable is specified for utility. As Fritz [1995] points out, a software program can in general be considered useful if it is used by the users, although that is not a sufficient nor a necessary condition.

A similar approach to evaluation was described by Eason [1993]. Eason defined eight independent variables which should be evaluated. Three of them refer to the system: *utility* (task match), *easiness of use* and *easiness to learn*. Three other variables refer to the user: *knowledge*, which includes experience using computers and related applications, *discretion*, which refers to the users' option to use the system, or whether they are forced to use it regardless of its usability; and *motivation* to use the tool. Finally, task characteristics include *frequency* of use, and *openness*, which is the degree of variability in the task requirements.

In order to measure the variables, there are two main kinds of experiments that can be performed: **field tests** and **controlled experiments**. In the first case, the system is actually used by final users, in a natural situation, and the opinions of the users are collected. In this case, the user actions have to be recorded either with automatic methods (by keeping a log file), or with an objective evaluator; and the user opinions are usually taken with forms and interviews. This is probably the best way to evaluate a system, because it is the final users who judge it, but there is the practical disadvantage that it is difficult to find a large set of final users willing to spend some time working with untried software.

Controlled experiments, on the other hand, have to be well designed in order to mimic real situations, but they have the advantage that it is easier to find volunteers. In these experiments, the users are asked to complete some tasks with the system, and next the variables are measured, with data such as the user actions, and questionnaires or interviews with the subjects.

9.4.1 Experiment

The evaluation of the research described has been done with a controlled experiment. The platform used as a server, to which the users could connect to use the system, was the same in which the off-line processing was performed. The three web sites were available to the users, but the experiment only included explicit tasks about two of them: Darwin's and Hegel's.

The hypotheses to be checked with the experiments were the following:

Hypothesis 1. The system is easy to learn by novice users.

Hypothesis 2. The filtering done with respect to the user interests is useful in order to access the more relevant information at once.

Question	Mean	Std deviation
Experience using a web browser	4.17	1.29
Experience using adaptive hypermedia	2.58	3.86
Fluency reading English	3.5	3.87
Previous knowledge about Darwin	2.08	3.3
Previous knowledge about Hegel	2.17	3.11

Table 9.6: Profiles of the users. Each question was evaluated from 1 to 5 (very low, low, medium, high and very high).

Hypothesis 3. The information automatically collected from the Internet is of interest to the user.

The procedures used in this evaluation are described in the following paragraphs.

User data

Before the experiment, the users that participated in the controlled evaluation were asked to fill in a form in order to learn the following user characteristics:

- Name.
- Sex and age.
- Previous experience using a web browser.
- Previous experience using adaptive hypermedia sites.
- Fluency reading English (the language in which the texts were written).
- Previous knowledge about Darwin's *The Voyages of the Beagle.*
- Previous knowledge about Hegel's *Lectures on the History of Philosophy.*

The other two variables described by Eason [1993] to model a user, discretion and motivation, had the same value for all the subjects, because the evaluation was a controlled one. Twelve people volunteered for the experiments, with different backgrounds: there were one linguist, three electrical engineers, one industrial engineer, two mathematicians and five computer scientists. Seven of them were males and five of them females. Nine of them worked in the department of Computer Science, some of them studying for their Ph.D. They were taught the different options of the system, and the profile of each person was collected.

Table 9.6 shows the characteristics of the users. In general, they all had much experience using a web browser (the standard deviation is very small), but only some of them knew what adaptive hypermedia is; fluency reading English was very variable, as some had a high proficiency, and others could only read slowly. Finally, a few of them knew about the texts used in the experiments, written by Darwin and Hegel.

Usability

The users were asked to play with the system until they felt confident that they understood all the functionality. In no case it took more than half an hour to experiment with all the functions of the system. At the end, a small questionnaire was given with the following questions:

Question	Mean	Std deviation
Easiness to register in the system	4.08	2.99
Did you find the texts interesting?	4.08	2.99
Was any stereotype similar to your interests?	3.91	2.98
Did you find useful the possibility of creating new stereotypes?	4.83	1.29
The length of the summaries is appropriate	4.18	1.91
The summaries are coherent and easy to read	3.42	2.63
The possibility of expanding the summaries is useful	4.67	2.58
The navigations options make it easy to find information	3.63	3.82
The annotations of links with colours is useful	4.42	2.22
The information collected from the Internet is useful	4.25	3.5

Table 9.7: Answers of the users. Each question was evaluated from 1 to 5 (very low, low, medium, high and very high).

- Easiness of registering into the system.
- Did you find interesting any of the processed texts? (Darwin, Osler, Hegel)
- Was any stereotype similar to your interests?
- Do you consider useful the possibility of defining your own stereotypes?
- The length of the summaries is right.
- The summaries are coherent and easy to read.
- Did you welcome the possibility of expanding the summaries to read the whole text?
- The navigation options make it easy to find the relevant information.
- The annotation of links with colours resulted useful.
- The information collected from the Internet is a helpful tool.

For each of these questions, each user answered with a number from 1 to 5, in order of increasing acceptability of the different options: very low, low, medium, high and very high. The answers are shown in Table 9.7. In general, most of the options were welcomed by the users, with a mean value between high and very high (greater than 4).

A specially good result was obtained by the possibility of creating the new stereotypes; however, one user commented that the procedure to do so should be made easier, as it took a long time to classify the one hundred paragraphs in two classes: relevant or irrelevant. Other system features that were praised are the annotation of links with colours and the possibility of expanding the summaries.

The reason why the possibility of expanding the summaries received a very high score is probably linked to the fact that the coherence of the summaries received the lowest score of all the features: 3.42. As stated in Section 8.2, one of the bigger problems of sentence extraction procedures is that the resulting summaries might not be coherent, because of pronouns and definite Noun Phrases whose antecedent is lost, dangling conjunctions, and other typical problems. This is probably the weakest point of the system.

Finally, the question of whether the navigation options make it easy to find information received the second lowest score, 3.63. Together with this question, some users provided useful comments, some of which are the following:

- At the top of the generated pages, there are two arrows that represent hyperlinks for direct guidance, linking consecutive sections from the original files. Several users noted that these arrows should also appear at the bottom of the document, so it is not necessary to scroll up in order to continue reading.

Question	Group 1		Group 2	
	Mean	Std deviation	Mean	Std deviation
Experience using a web browser	4	0	4.33	1.15
Experience using adaptive hypermedia	2.83	3.29	2.33	1.83
Fluency reading English	3.33	3.06	3.67	2.31
Previous knowledge about Darwin	1.83	2.2	2.33	2.31
Previous knowledge about Hegel	1.83	2.61	2.5	1.22
Reading speed	169	64.33	189.91	166.85
Reading efficiency	121.64	90.89	124.06	187.54

Table 9.8: Profiles of the users. Each question was evaluated from 1 to 5 (very low, low, medium, high and very high).

- The system should keep track of the pages that have already been visited by the users, and allow them to go back to them easily.
- Some users wanted better search procedures that allowed them to perform a search on the whole adaptive web site simultaneously. Given that every single page is generated on-the-fly according to the user profile, it is not a trivial task to do.

In addition, as some users pointed out, the speed of the system can be improved. This is something that can be worked out because, at the moment, the XML library being used requires that whole files have to be read into memory in order to use some part of them. A better XML library that could index the files and read only the required portions would improve very much the performance. Secondly, there is always the possibility of using a better machine with more memory.

Utility

In order to measure utility, the users were divided into two groups, as homogeneous as possible. The users were also asked to perform the reading efficiency test, and special care was taken in that the two groups contained a similar distribution of this characteristic. Table 9.8 shows the distributions of the properties of the users from the two groups. Both have a reading efficiency of around 120 words per minute (see the last row in the table).

The users were asked to complete two different tasks. Firstly, in eight minutes, to collect all the names of animals found in *The Voyages of the Beagle*. Secondly, in five minutes, to collect all the biographies found in the *Lectures on the History of Philosophy*. The users from the first group, who were able to use WELKIN, created a profile for each of the texts, using the stereotypes *biology* and *biography*, respectively, and browsed the text from section to section collecting the requested information. They were able to find an average of 19.17 animals, but there were big differences, ranging from the 10 to 34 animals. They also found an average of 10.6 biographies, with values ranging from 10 to 12.

On the other hand, the users from the second group were given a text editor (`emacs` for the Linux users and `Notepad` for the Windows users) open with the text, and were told that they could use any of the functionalities of the editor. Concerning the animals, most of the users did not find any useful option of the editor, and limited themselves to reading the text and marking the animals they found. They got a mean value of 15 animals, with individuals scores ranging from 9 to 23. Concerning biographies, most of the users spent half of the time thinking what to do or browsing the text purposelessly, until they found by chance

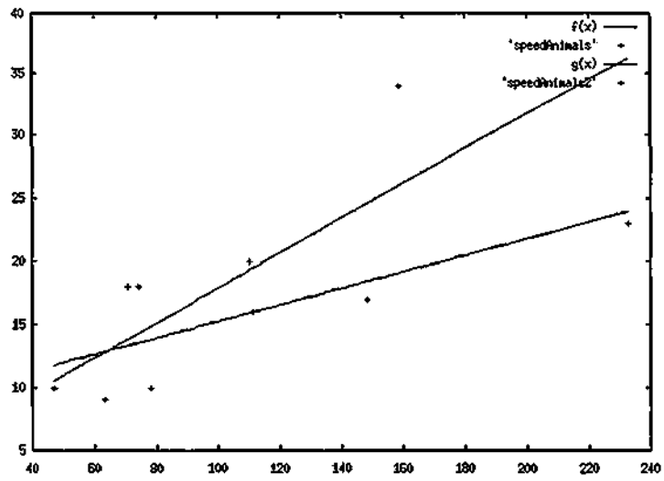


Figure 9.7: Number of animals found in function of the reading efficiency in English. The top line represents the performance of the users with WELKIN, and the line below the performance of the users with a standard text editor.

a biography, and next made a search through the document for the word *born*, which allowed them to find most of the information. They found a mean of 7.6 biographies, with individual scores ranging from 6 to 9.

It can be expected that the number of results provided by each user is related to their reading efficiency: the faster a user reads, the more animals he or she finds. Figures 9.7 and 9.8 display the linear regressions between the reading efficiency and the number of results found, for each of the groups. In both of them, it can be seen that there is correlation between both values, as all the regression lines are increasing.

The mean results of the users with WELKIN are all higher than the results of the other group. Therefore, it is natural to test whether we can statistically prove that WELKIN offers a better performance than standard text editors. Given the size of the sets of users, and the large standard deviations in the results, a hypothesis test shows that the difference between the animals found by the two groups should have been of at least 16.6 if we wanted to affirm with 95% confidence that the number of animals found with WELKIN is higher than the number found without it. Because this difference happens to be 4.17, we cannot conclude anything. And the same happens if we consider the biographies.

However, the fact that the mean number of animals and biographies found with WELKIN is higher means that it is very possible that performance increased. It would be necessary to do the experiments with a larger population, in order to prove it. It is also worth to note that the users that used WELKIN had to read a smaller amount of text, as some of the time was spent by the system generating the pages. That is also an important variable if we are to consider the fatigue of the users.

User satisfaction

Finally, the users were asked to answer a final question about their general satisfaction after using the system. This question obtained a mean value of 4.17, with a standard deviation of 1.29, a value which shows that they all agreed in that they had liked the system. Everyone asked either 'high' or 'very high'.

To end with the evaluation, every action performed by a user was written in a log file. Because all the values obtained so far have been obtained from controlled experiments and with few users, it is difficult

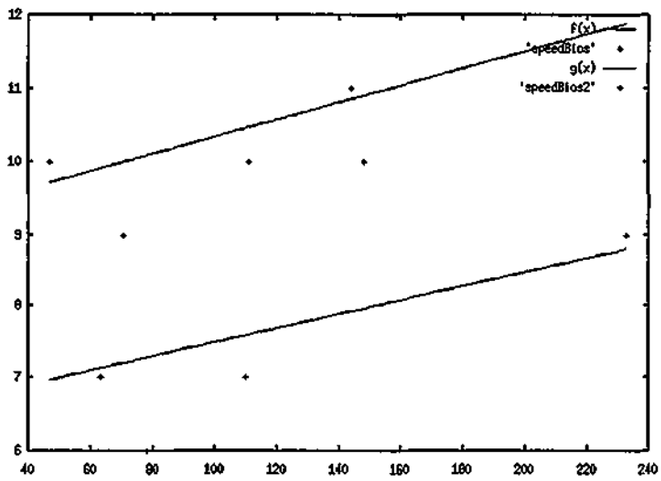


Figure 9.8: Number of biographies found in function of the reading efficiency in English. The top line represents the performance of the users with WELKIN, and the line below the performance of the users with a standard text editor.

to extract conclusions from them, but we plan to apply this feature to obtaining data about how possible users that access outside a controlled experiment behave in the system. Section 10.3, about the future work, discusses this.

9.5 Summary and discussion

The application of the system to three different texts shows that it produces a similar amount of errors, a fact from which we can conclude that it can be easily applied to different topics, provided that they contain the kinds of terms that it is able to tackle: dates, locations, people, animals, bodies of water, etc. It would probably need some adjustments if it were to be applied to other domains, such as technical documents or texts dealing with abstract terminology.

The evaluation with the users show that it has been well accepted, and most of its features have been praised. The main drawback the users found is the quality of the generated summaries, which sometimes produces incoherent, but the navigation options and the filtering were welcome by all the users.

Chapter 10

Conclusions and Future Work

The work described in this thesis presents several contributions to the field of Adaptive Hypermedia. Furthermore, it describes several advances in some of the collateral techniques that have been used, which include lexical ontologies, text summarisation and accurate searches on the Internet, amongst others. This chapter describes the contributions of the work, and possible ways in which it can be further improved.

10.1 Contributions

An overall integration of diverse approaches

The strong point of this work is the procedure in which different algorithms, components and techniques have been integrated to fulfil one single goal. Techniques borrowed from the field of Natural Language Processing have been applied to knowledge acquisition and text generation; challenges from the Semantic Web community have been addressed for the automatic collection of accurate information from the Internet; and different adaptive hypermedia methods and techniques take advantage of the output of the previous modules in order to show tailored information to different users. The architecture integrates more than fifteen components written in different programming languages (C, C++, flex, prolog, Java, shell script and Javascript), which interchange information via an XML encoding scheme, in a way that all the information generated by every module is used, in a way or other, by some other module.

A new architecture

A new architecture has been described for automatically transforming linear texts into adaptive hypermedia sites. It incorporates text processing techniques for analysing linear texts and extracting information from them. Text understanding is performed in a shallow way, learning the semantics of unknown concepts by classifying them inside the lexical semantic network WordNet. By combining hypermedia generation with user modelling techniques, the text provided by the system is adapted to the user's interests. New ways have been devised in which the interests of users can be represented, in a fine-grain way that allows the encoding of small variations in the user's interests.

The system has been implemented and tested. It has been applied to different texts and topics, with good results. It has also been evaluated in a controlled experiment, where it was very well received by the

users. The experiments suggest that productivity increases when performing certain text analysis tasks, and the users found the system usable and appealing.

Automatically constructing a hypermedia site can be done in 60-90 minutes. If special stereotypes have to be defined for the site, then it may take between one or two hours more because the texts have to be annotated in order to train the topic filtering module. This means that a complete hypermedia site can be obtained from a linear text quickly, and a whole collection of adaptive hypermedia sites can be available for use in just a few days. Compared to the manual work of creating a web site from an electronic text by hand, or using standard text and hypertext editors, it highly improves the productivity, apart from offering the term identification and classification results, the adaptation and summarisation facilities, and the various navigation options.

The modular architecture allows a module to be easily replaced by other, with no alteration to the remaining portions. In fact, some of the modules, such as the Term Classification component or some of the algorithms for linguistic processing, were changed as the algorithms evolved, and their replacement did not alter the overall functioning of the system. This will also make it very easy, in the future, to incorporate one or several of the possible improvements that have been identified and are described below in Section 10.4, which examines ongoing and future work about the separate modules.

Accurate Internet searches

An optional module has been built that performs the searches on Internet, and allows the user to collect high-precision additional information about the relevant concepts. This was one of the features more appreciated by the users, and offers many possibilities. If someone needs to gather relevant information about some domain-specific concepts, and there is one or several documents about them, it is possible to use it already to collect information from the Internet, so the searches will be performed and very accurate information will be collected automatically. In this way, the system can be used not only as an adaptive hypermedia site generator, but as an Information Retrieval program, running on top of the web search engine `Google` or any other, for finding accurately relevant information in the Internet.

Components

This work also describes new algorithms for each of the components of the system, some of which have applications for many other systems and architectures apart from hypermedia generation. Each of these components has been evaluated independently, and it has been seen that their accuracy rivals other of the current approaches. For one of them, the *Term Classification* module, a benchmarking set has been proposed, and it has been made public for others to use as well.

Section 10.4 contains a separate discussion on each of the individual components, and describes future work that can be addressed to improve each of them.

10.2 Comparison with other existing approaches

The work described in this thesis combines ideas from User Modelling, Hypermedia, text to hypertext conversion and Natural Language Processing. Therefore, the different algorithms and design issues in the system should be compared with the individual approaches that address each of them separately. This has

been done throughout the thesis. This section will focus only on approaches that refer to the top-level goal: automatically generating hypermedia sites.

Concerning text to hypertext conversion, Blustein [1994] distinguishes three different kinds of systems. Firstly, there are those that create a site from **a highly structured text**, such as dictionary definitions [Raymond and Tompa, 1988]; or those that expect a set of **layout annotations** inside the linear text from which the structure of the generated hypermedia site can be guessed [Furua et al., 1989]. With these procedures, the information that is taken into consideration consists of the sections of the documents, and the explicit references written in some markup language, such as *roff or LaTeX, to figures, tables, footnotes, references and to other sections. One example is the LaTeX2HTML, a widely used command that changes a LaTeX document into a set of hypertext nodes with automatically generated hyperlinks between them, including direct guidance to the previous and the next section, and index pages.

These approaches for text-to-hypertext generation usually start by segmenting the text into smaller units. Markup inside the document specifies the portions in which the text has to be divided. When there is no markup but the formatting guidelines of the document are known, it may be also possible to identify section titles and paragraph boundaries automatically. In the particular case of hypertext dictionaries and encyclopaedias, there are usually abbreviations indicating the function of the different sentences in each entry.

WELKIN has in common with these approaches that it is fully automatic; the only needed resource is the original linear texts, and a *script* performs all the processing with no user supervision. WELKIN provides the following additional advantages:

- A linguistic processing of the texts allows the acquisition of some knowledge from them, such as the classification of high-frequency words in an ontology.
- There is no need of markup in the text, such as LaTeXor SGML commands.
- The final output is generated dynamically for each kind of user. On the other hand, it is still possible to read the texts without any adaptation, by registering a user interested in everything and without compression, for whom all the information will be available, and nothing will be filtered out or summarised.

There are other methods for text segmentation that do not require markup [Hearst, 1993]. These methods look for topical units, i.e., portions of the text that refer to the same topic. WELKIN combines both approaches, as

- Section and paragraph boundaries are automatically found using layout clues, such as indentation, underlined sentences, sentences in uppercase and keywords (e.g. *prologue*, *introduction* or *chapter*).
- The topical units are identified at the topic filtering step, without the need of any markup in the text. With WELKIN's approach, furthermore, it is possible to classify a portion of the text as pertaining to several topics at the same time, something that occurs rather often in the texts, as there may be overlapping between several topics.

A second approach for hypertext generation makes use of AI techniques to transform text into hypertext. Two of the earliest systems are VISAR [Clitherow et al., 1989] and TOPIC [Hahn and Reimer, 1988]. The first one is used for maintaining a database of journal citations, and the second one represents the topical

structure of a text as a hierarchical graph, with relationships between the portions on the text. Both use linguistic processing to some degree, and are usable only on specific domains, because they make extensive use of domain-specific knowledge bases, so they are not easily portable for different kinds of texts.

Compared to these approaches, WELKIN has in common that it also performs a linguistic processing of the texts and uses the lexical knowledge base WordNet. However, the main difference is that WordNet is a general-purpose knowledge base, not domain-specific, and at the first steps of the processing it is tuned, by acquiring new concepts, to the specific domain treated in the text. Therefore, it has been successfully applied to texts on biology, history, medicine or philosophy, and it has proven able to handle unrestricted documents collected from Internet.

Finally, there is a third approach that can also be considered a text-to-hypertext approach, and it is applied when there are several separate documents, and Information Retrieval techniques are used in order to group the related documents, and to add links between the ones that are related in some way. IR tools are very useful for this task [Blustein, 1994], as techniques such as the vector model can be used to calculate similarities between the different texts. That is the approach used by WELKIN for the topic filtering, as vectors of words have been used to perform the topic filtering to calculate the adequacy of paragraphs and summary sentences to the user.

In summary, the three different techniques have been applied to different problems found during the design of WELKIN, and as can be seen, they can be integrated in a single system, by applying each to the aspect for which it is more useful.

There are few works that address the topic of transforming linear texts into adaptive hypermedia. The following paragraphs describe the differences between the framework described in this thesis and other related works.

There are other systems that use text generation techniques for hypermedia generation. Oberlander et al. [1998] and Milosavljevic et al. [1998] describe two approaches for creating virtual museum guides and adaptive encyclopaedias, and these have the advantage that they use a complete Natural Language Generation component. Thence, the generated text is more flexible than extracting prewritten sentences, because the sentences themselves may be adapted to the user's interests and previous knowledge.

On the other hand, using a full Natural Language Generation approach has the problem that creating the Knowledge Base is typically very time-consuming. Oberlander et al., and Milosavljevic et al., used semiautomatic procedures for acquiring information from semi-structured data; for example, Milosavljevic et al. [1998] used a museum database where some of the information, such as the date or location of an item, was stored as separate fields in the entry; and only attempted some simple extraction techniques for the fields which contained unrestricted text. By using a summarisation technique, WELKIN loses some flexibility, but succeeds in generating complete sites from unrestricted documents.

The same applies to DiMarco et al. [1997], which describes a system for providing medical advice. The output shown to the user is produced with NLG techniques, using *master documents* which contain the information to be shown to users with different profiles. In order to acquire these master documents, the system uses extensively NLP tools such as a semi-automatic parser, a coreference solver and a rhetorical parser, but ultimately the documents are written by "a professional technical writer or Web-document designer". Although full NLG allows the system to adapt the output in a more sophisticated way to the user, our fully automated approach based on summarisation frees the web designer from the knowledge acquisition bottleneck.

Ragetli [2001] describes a procedure for structuring a set of pages in a hypermedia structure, by creating

a concepts hierarchy and next linking the documents to the concepts. However, some differences that we may cite are the following:

- The hierarchy is constructed manually from terms obtained from a glossary, a process that involves much work. Some methods are described that could help in this process, but they are not evaluated. For example, Ragetli points out that some of the concepts are classified using lexical matching [Neville-Manning et al., 1999]; in this way, the relationships between *air transport* and *transport*, or between *conditional probability* and *probability* are produced. It is a good idea, but which must still be refined, as no explanation is done on how complex terms such as *conditional probability* are distinguished from modified words that are not terms, such as *happy boy*. Term Identification procedures [Vivaldi, 2002] would probably be of great help here.
- The documents are linked to the concepts using IR techniques based on the vector space model.

There are several differences between this work and WELKIN, a few of which are the following:

- In WELKIN, the internal structure of the documents is analysed, and the generated pages are obtained from portions of the source documents. Some of the generated pages have text fragments that are not consecutive in the original text collection. In Ragetli's approach, the hypermedia pages are the complete documents.
- In WELKIN, there is a user model, and the contents shown to the user are dependent on the user's profile. Even the structure of the site is dynamic, depending on the user model.
- Finally, Ragetli [2001] describes an unfinished work, that has yet to be fully implemented and tested.

10.3 Future work

The following are some ways in which this work can be further improved, or applied to different problems:

- By acquiring information not only from electronic texts, but also from existing static hypertext, it would be possible to create a system that produces an adaptive hypermedia site from a combination of texts and existing web sites. This can be useful for generating personalised views of intranets or web sites, by showing only to each user the information that will be relevant, and omitting the rest. All the links to irrelevant web pages could also be suppressed, so the web site would be reduced only to the relevant pages and the user would get lost less likely.
- In line with the previous comment, another useful improvement would be the addition of a search engine that looks in all the relevant sections of the web site. With the current architecture it is not a trivial task, considering that the web pages are generated on-the-fly, so it would be necessary to generate them all each time the users' interests change in order to perform the search on the results. A possible solution would be to perform a search on the whole web site, and next filter the list of hits, so it is not necessary to perform the adaptation on the whole web site.
- It would also be useful to create some system for exporting the results. For example, someone might be interested in creating a static hypermedia site on *The Voyages of the Beagle* from the point of view of biology, with no concern on the adaptability to the users. In this case, the system could easily generate all the possible pages for the stereotype, and write the result in HTML format as a static web site.

- The usage evaluation has been done, till now, in a controlled experiment, but it would be very interesting to repeat it with a larger pool of users that need to use the system in their work or for some other reason, so they spend much time working with the system. Also, if this experiment was performed, the log files that record all the user actions could be analysed to extract more conclusions about the most popular actions performed by the users, the weak points of the system, the actions that are slowest to execute, etc.

10.4 Discussion and Future Work of the Components

The architecture described is divided into several components, each of which can be implemented internally in many ways. The advantages and limitations of the approaches with which they have been designed and implemented have already been described previously, when each of them were evaluated. This section lists the more relevant components in the design, and enumerates future lines for research on each one of them. Some of the research lines are complex enough for large research projects centred on them alone.

10.4.1 Term Classification

The work presented here for Term Classification is original in the sense that it is, to my knowledge, the first approach fully unsupervised for extending a lexical ontology with new terms. Previous approaches either used hand-coded heuristics, required a manual supervision by a judge, or included rules or scripts that had to be coded before using the system. With the new approach presented here, only the ontology and a set of domain-specific documents is necessary, and all the processing is done automatically by examining automatically collected corpora.

Most of the limitations of the current approach have already been described in the thesis. The most important weak points are the following:

- Only one hyperonym is provided for each unknown term. It might be desirable, for example, to learn that *Chiloe* is both an island and an administrative district.
- The classification is very dependant on the contexts in which the words appears, and it is hence necessary to see the word in context a certain number of times for the classification to be accurate. Low-frequently terms will not be properly classified using this technique.
- Some fine-grained distinctions are not easily distinguished by the contexts. Terms that are semantically very distinct, such as a location, a person or a river, will undoubtedly appear in different contexts; however, for words that are very semantically related there is much overlapping of the sets of contexts in which both can appear, and a limited corpus might not reflect the differences. This is the case of the distinction between men an women, who were very difficult to make for the algorithm in the experiments.

Several improvements can be made in order to increase the accuracy for these particular cases:

- The algorithm may be implemented as a beam search, where different paths are explored at the same time. This modification might be desirable for two reasons: firstly, if the algorithms makes any incorrect decision right now, it is impossible for it to amend it, because it only proceeds downward. With a beam search, it would be possible to explore several paths at the same time, and only at the end it would

discriminate between the synsets at which each search path ended. Secondly, it is also possible to implement it so in the end the output is a ranked list of synsets that were judged possible hyperonyms, and the new synset could be attached to more than one hyperonym.

- Other open line for future research consists on acquiring signatures for every synset in WordNet, including the remaining entities and other kinds of concepts, such as *psychological features* or *groupings*. This has not been done till now because of limitations on network and CPU speed, and storage requirements, but hopefully as the hardware improves it will be possible to collect them all.

- The design of WordNet has been sometimes criticised. Some of the common critiques to WordNet are that there are less hyperonymy links than there could be, and that it is too fine-grained, distinguishing senses of words whose meaning is nearly the same. It might be worth to check whether this approach also works with different ontologies, such as the upper level Cyc ontology [Lenat, 1995].

- Classification accuracy could be improved using other features other than contextual signatures. For example, by observing Table 9.1, it is seen at first sight that several terms that started with the word *Mr.* were classified as women. Simple heuristics such as the knowledge that some personal titles only refer to the male gender, combined with a gender annotation of WordNet, could lead to a proper classification of these synsets. In the same way, WordNet could be extended with other kinds of constraints in order to block some search paths because the gender, number or any other kind of feature of a synset has a different value than the term's. This is specially relevant for the cases in which there are few contextual evidence about an unknown concept.

- Concerning the signatures, other kinds of syntactic relationships can be explored. Furthermore, a generalisation of the words in the signature, if performed properly, might increase the classification accuracy. A possible way to do it are the procedures by Hearst and Schutze [1993] or the fuser concepts introduced by Hovy and Lin [1999].

- Agirre et al. [2001] notes that some filters on the downloaded documents from which the topic signatures are extracted, such as choosing only one document from each web site domain, may improve their quality. That is a field of research that might be worth studying.

- Finally, the problem of finding whether a word is being used with two meanings in a text is very hard (e.g. *St. Jago* meaning a person and a location in the same document). A partial solution might consist in using techniques for low-frequency terms, such as Named Entity recognition procedures, and study whether the same word is classified as different entities in the same text. If that happens, that term has to be studied in more detail by the system, in order to classify each appearance of the term correctly.

  In the same line lies the problem of finding whether a term found in a text that is already in WordNet is being used with one of the known senses, or whether it is used with a different new meaning.

As a final remark, this kind of processing could also be applied for clustering concepts (using the similarity metric between topic signatures) to create an ontology from scratch; or for studying the ways in which the use of a word in context changes in different centuries, just to cite a few examples.

10.4.2 Distributional Semantics

Regarding the Distributional Semantics hypothesis, although it has been applied to many problems, there are few works to prove, either theoretically or empirically, its validity. One of these is the work by Levin [1993], which is a good study on the distributional properties of English verbs.

Section 4.3 describes a way in which several concepts are taken from WordNet and a correlation is found between the distance between their semantic meanings (using WordNet to measure it) and the distance between the contexts in which they appear. However, it would be interesting to generalise this work by selecting much more sets of WordNet synsets randomly, and checking with statistical tests that the same applies to any set of synsets randomly chosen.

It might be the case that for particular portions of WordNet the correlation is larger, and that for other portions it is smaller. It can be expected, from the results of the Term Classification algorithm, that some semantic distinctions, such as that between men and women, are more difficult to capture with context words, so it is presumable that interesting conclusions will be drawn from these experiments. Finally, the same experiments can be performed with words that are inside the same synset, to test whether they are really synonyms or plesionyms.

It is important to note that, for these experiments, special care must be taken that the documents collected from the Internet in order to collect the contexts have a good quality. It might be useful to use the filtering techniques described in the previous section for collecting Internet data, and to explore new ways to ensure that the words are used in the documents with the correct sense.

Concerning the different kinds of topic signatures, a preliminary test was performed to see whether a generalisation of the words using WordNet could improve the classification results (see Section 8.1.1, *Topic filtering*). As already indicated there, a better selection of the synsets which should be generalised might produce better results. Li and Abe [1997] and Hearst and Schutze [1993] describes two possible ways in which it might be performed.

10.4.3 Analysis of temporal expressions

Concerning the identification and interpretation of the temporal expressions in the texts, there are still several improvements that can be performed:

- Firstly, the evaluation has been done with a small benchmark corpus, and it would be desirable to repeat it with a larger test data. Setzer and Gaizauskas [2001] describes a study in which different human annotators marked the events and temporal relations found in a text corpus, but he notes that the inter-annotator agreement was low, a fact that means that the guidelines for annotating temporal information in texts still have to be made clearer.

- It is also important to detect event's coreference, so two references to the same event are recognised.

- Some modules, such as the word-sense disambiguator or the parser, are the origin of many of the errors in the first stages, and should be improved.

- People's ages can also be considered as date indicators.

- Extend the framework to represent fuzzy intervals where the start or end time are not clearly specified. For instance, expressions such as *one year ago* can refer to an undetermined point inside a period of

time that is left very indeterminate. Further information in the text might later restrict the period to which that expression referred, for instance, indicating a particular day.

10.4.4 Summarisation

This work includes a new approach for text summarisation based on generating extracts of texts with genetic algorithms. The method is easy to program, so it can be easily be redone for other purposes.

As stated before in Section 7.2.1, most procedures for generating a summary by using extraction have one important drawback in that they do not take into account the compression rate for choosing the sentences. With the example used before, if the top two sentences, s_1 and s_2, provide together an idea; and the third sentence in the ranking, s_3, provides alone other important idea, then a 1-sentence summary should select s_3, because it includes a complete idea, than either s_1 or s_2 alone. In other words, the weight of the extract should take into consideration all the sentences that have been selected and the relationships between them.

This is also specially relevant for multi-document summarisation. If there are two sentences selected from different documents that share the same meaning, then the summary should receive a lower weight than other summary that contains one of those sentences and a different one.

Another feature that is easily programmed with this summarisation approach is the ability to generate user-oriented or viewpoint-oriented summaries. Just by adding to the fitness function a component that takes into account the adequacy of the sentences or groups of sentences to the user's requirements it is possible to tune the generated summaries to the users' interests and goals.

With the genetic algorithm, all the constraints on the target summaries can be easily solved. The important design decisions now touch on the fitness function, but measures of sentence cohesion and meaning overlapping can be easily encoded in this function, which will guide the evolution of the population of summaries. Genetic algorithms also have the advantage that the search performed is nonlinear, but they can be programmed so the performance is not slow, and they can be built for practical applications.

As a consequence of this, most of the future work respecting summarisation relates to this fitness function. In the approach built and tested, several characteristics of the sentence that were deemed distinctive of a good summary were encoded in the fitness function, but there are still other features that could be encoded in it: sentence meaning overlapping; the cohesion between one sentence and the others in the summary; etc.

The genetic algorithm used is a basic one, but there are many different possibilities for improving it and guiding it to the correct solution more quickly, such as using more sophisticated ways of performing the crossover between the genotypes, or starting with smaller summaries and make them grow as the best populations are being found. Using local search around the selected summary after the genetic algorithm in order to finetune the resulting summary also produces good results.

The proposed fitness function is only experimental, and most of the future research we are planning concerns it. Apart from introducing new heuristics, we are aware that a linear combination of the different values might not be the best solution, so other possibilities should be explored. The approach presented here is unsupervised, but a future objective is to make the fitness function evolve as well as the summaries, in a supervised environment.

Note that the algorithm does not restrict the length of the summary. We can easily define the genotype as a boolean vector with zeros and ones, and define that the sentences extracted are those such that the corresponding gene has a value of 1. In this case, we can *evolve* a population of summaries whose length is not fixed beforehand.

Finally, the summarisation procedure should be extended with some kind of linguistic processing, such as resolving the antecedents of the pronouns, so that a pronoun whose antecedent has been removed can be replaced by it. Linguistic processing might also be useful for merging the resulting summary sentences in order to produce a more condensed output.

This algorithm has been used to participate in the DUC-2003 competition.

10.4.5 User modelling

The user modelling component allows the users to specify their interests and the level of compression to be performed on the texts. However, there are many ways in which this can be improved:

- Currently, there are two means of producing user-adapted profiles. The first one, consisting in annotating one hundred paragraphs by hand to train the classifier, is a slow process, as one hundred paragraphs have to be read and classified one by one. The second one consists in declare interest for everything, and proceed discarding paragraphs while browsing the site. It could be interesting to investigate other means of indicating one's interests for the topic filtering.

- By modelling the user's previous knowledge, it is possible not to present the same information to an user twice, and the system can take advantage of the facts that the user already knows in order to add additional explanations comparing the current topics with other information that the user has already visited.

- Individual traits such as age or language could also be included in the user model. This, combined with a language generation procedure, would allow the contents to be presented in different language styles.

- In principle, the user model could be extended with any kinds of information, such as psychological traits, hardware requirements, etc.

- The current design of the generated web sites, with three frames on the screen, can be seen nicely on screens with a resolution of 800×600 or higher. However, it is not user-friendly on smaller resolutions, as it is necessary to scroll the screen, both up and down and left and right, to see all the contents of the pages. If the hardware and software settings of the user were modelled, the presentation could be adapted to them. Some users have also suggested that using clearer colours will improve the user interface.

10.4.6 Collection of documents from Internet

With respect to the collection of additional information from Internet, Section 8.4 describes a novel way to collect information about the relevant terms, trying to ensure as much as possible the quality of the obtained information. The filters that have been used, such as the use of the *concept signature* or the heuristic of looking for the concept in subject position in order to ensure that the information is relevant about it, are new and helped eliminate most of the irrelevant data.

The generated summaries still can be further improved. The following are some ideas that can help for improving their quality:

- A reordering of the paragraphs inside the summary document according to their topic. Some of the collected paragraphs may refer about the history of a city, while others may refer about its economy, its population and folklore, etc. The paragraphs that refer to the same topic should be put together in the summary, in order to improve the coherence.

- A merging of the paragraphs that convey the same information. Indeed, many of the generated summaries contain a paragraph repeated several times, because it appeared in more than one document. Sometimes there it a whole paragraph that is nearly identical, except for one or two words, or for the punctuation symbols. In these cases, the repeated information should be discarded.

  A more complex case happens when two paragraphs have overlapping information. In this case, it would be desirable to merge them together in one single paragraph that provides all the information, using natural understanding and NLG techniques.

- Identification of contradictions. In some of the generated summaries there were paragraphs that provided contradictory information, such as different dates for the same events. It would be desirable to be able to identify these, and to provide the source from which every material was obtained, so the user can judge which is the one that merits belief.

Finally, this procedure can be extended with `Google`'s facility of searching images on the Internet, so images relevant to the terms are also collected. Additionally, with image analysis techniques it is possible to determine whether an image is a photograph, or a diagram (e.g. a map), so the images can be further classified and shown depending on the user interests and needs.

Postscript

Artificial Intelligence has often been considered a bag of odds and ends containing the study of many different problems and techniques. We can find logic formalisms, knowledge representation languages, problem solving methods, planning algorithms, natural language processing applications, case-based reasoning, spacio-temporal reasoning, multi-agent systems, the semantic web and ontology representation formalisms, and many other fields of study, all classified under the same label, AI.

It is difficult to provide a definition acceptable to everyone, but most will agree that the design of AI is the construction of machines that can mimic the cognitive actions and reasoning abilities that are characteristic of humans. Shop assistants are expected to remember people tastes and shopping history, and to behave like a small shop attendant does; planners are expected to schedule the several tasks and to book the necessary resources as a human would do; and dialogue interfaces are expected to speak as a real conversational partner.

It is my feeling that, although many of these problems are studied separately, when they are ripe they converge, and that enriches the research and paves the way for other improvements. The examples are many. Hypermedia together with User Modelling stimulated the appearance of Adaptive Hypermedia; Hypertext and Knowledge Representation are the starting point of the semantic web; Natural Language Processing benefits both from theories from linguistics and psychology, and is being increasingly applied to other fields; and complex dialogue interfaces have to deal with language recognition and generation modules, text analysis, emotion recognition and simulation, and knowledge representation in order to create sophisticated products.

In conclusion, it is equally important to advance in one field than to be able to blend the different advances into one single architecture. All of the previously mentioned fields (planning, problem solving, natural language processing, spacio-temporal reasoning, etc.) are abilities that humans, with a proper training, can perform. If the automation of these abilities improves, and proper algorithms are devised to simulate them all with human performance, and they are merged smoothly into one single architecture, mayhap some kind of Artificial Intelligence will arise, posing an array of metaphysical questions about human nature.

Thus, it is important that the different fields have not closed boundaries; interdisciplinary collaboration is essential for progress. As the separate fields advance, their cooperation may result more and more fruitful.

Appendix A

Abbreviations

AH Adaptive Hypermedia

AI Artificial Intelligence

DR Document Routing

DS Distributional Semantics

DUC Document Understanding Conference

FOPL First-Order Predicate Logics

GNE General Named Entity

GORT-4 Gray Oral Reading Test, 4rd Edition

KA Knowledge Acquisition

KB Knowledge Base

KE Knowledge Elicitation

HCI Human Computer Interaction

IC Information Content

IE Information Extraction

ILI Inter-lingual Index (in EuroWordNet)

ILP Inductive Logic Programming

IR Information Retrieval

LA Learning Accuracy (OR metric)

LOTR The Lord of the Rings corpus

LKA Lexical Knowledge Acquisition

LKB Lexical Knowledge Base

MDS Multi-document Summarisation

ME Maximum Entropy

MRD Machine-Readable Dictionary

MT Machine Translation

MUC Message Understanding Conference(s)

NE Named Entity

NLG Natural Language Generation

NLP Natural Language Processing

OR Ontology Refinement

PoS Part-of-speech

QA Question Answering

RST Rhetorical Structure Theory

SAT-9 Stanford Achievement Test, 9th edition

TE Term Extraction

TREC Text REtrieval Conference(s)

WJ-III Woodstock-Johnson III Test of Achievement.

WN WordNet

WSD Word-Sense Disambiguation

WSJ Wall Street Journal corpus

WWW World Wide Web

Appendix B

Engineering Work

Before analysing the texts for classifying the paragraphs or identifying relevant terminology, it was necessary to perform some linguistic processing. This includes identifying word and sentence boundaries, morphological analyses and some shallow parsing. Many of the techniques described in this chapter have already been used before, or they consist of hand-coded heuristics and very specific algorithms. These tools have been extensively used in this work, and they are referred often in the description of the different modules. Section B.1 describes these tools.

Also, there were some characteristics of WordNet that had to be changed, in order to allow the programs to add new synsets to the semantic network, and to be able to apply the classification algorithms on it. Section 5.4.1, *Partition of instances and concepts*, described how new terms acquired with a Term Identification procedure might be classified as instances and concepts. Section B.2 extends the topic, indicating the definition for *instance* and *concept* used in this work and the reasons why it was adopted, and how the original WordNet was extended with this information. The section ends with some additional changes that were performed to the lexical network.

B.1 Linguistic tools

B.1.1 Segmentation

The segmentation is the first step that is performed on a text, with a double aim: a *tokeniser* finds word boundaries, and a *sentence splitter* locates the sentence boundaries.

The tokeniser has been programmed as a list of regular expressions for defining the different tokens, such as words, numbers or punctuation symbols. It is written in flex, which is fast, simple and portable. We allow tokens to contain letters, numbers, hyphens, slashes and dots. A quote is allowed inside a word-token if it appears in the second position, and the first letter is either an 'O' or a 'D', in order to account for some proper names. The tokeniser also expands abbreviations such as `n't` and `'re` into the whole words (`not` and `are`), when they are not ambiguous.

The last version of the sentence splitter follows (although not strictly) the design described by Mikheev [2002]. First, all words that end with a point are studied to decide whether they are abbreviations or not:

- If they are followed by a non-capital word, they are abbreviations.

- If they are followed by a capital word, but they were seen somewhere else in the document followed by a non-capital word, they are abbreviations.
- If none of the previous rules hold, but they look like abbreviations (e.g. a sequence of capital letters separated by dots) they are abbreviations.
- Otherwise, they are classified as non-abbreviations.

Secondly, the information about which words are abbreviations is used to identify the sentence boundaries:

- If the word that ended with a point is not an abbreviation, it marks a sentence boundary.
- If it is followed by a non-capitalised word, it is not a sentence end.
- If the abbreviation appears in a list of abbreviations that must be followed by a proper noun (such as title designations), then that is not a sentence boundary.
- If none of the previous holds, the rest of the document is scanned looking for that same word, to study other appearances of it, in order to decide which is the best decision to take. If the words involved were seen often, then it is assumed that it is not a sentence boundary. For example, in the text `U.S. Congress`, the first word is recognised as an abbreviation, and it is followed by an uppercase word. But because this combination is likely to appear again in the same document, it will not be marked as a sentence boundary.

B.1.2 Part-of-speech tagger

A part-of-speech tagger labels every token in the text with each part-of-speech. We are using the part-of-speech labels from the Penn Treebank [Marcus et al., 1993], some of which are represented in Table B.1. A part-of-speech (PoS) tagger usually takes into consideration the context of a word in order to decide how it should be labelled, either with transformation lists [Brill, 1995], a maximum entropy approach [Ratnaparkhi, 1998] or Markov chains, amongst other possible procedures.

WELKIN uses the TnT part-of-speech tagger [Brants, 2000], which is a probabilistic tagger based on the Viterbi algorithm. Trained on the Penn Treebank corpus [Marcus et al., 1993], it has an accuracy of 97%, which is one of the best scores among freely available taggers.

B.1.3 Morphological analyser

The purpose of a morphological analyser is to study the structure within words, by identifying the *root* or *stem*, which contains the basic meaning of the word, and the *bound morphemes* (prefixes and suffixes), which vary these basic meanings, for instance, by pluralising a noun (e.g. *parent* and *parents*), or by changing an adjective into a noun (e.g. *wide* and *width*).

The system does not use a full morphological analyser, but only a stemmer that obtains the root of the nouns and the verbs, without providing any other information (number, tense, person, etc.) The stemmer is basically a list of flex rules to recognise the ending inflections of English nouns and verbs, and uses an extensive list of exceptions that has been collected from WordNet, the Susanne corpus [Sampson, 1995], and from the morphological analyser in the LaSIE IE system [Gaizauskas et al., 1995], which was freely available.

part-of-speech	morphological variation	tag
noun	singular	NN
	plural	NNS
	proper, singular	NNP
	proper, plural	NNPS
adjective	normal	JJ
	comparative	JJR
	superlative	JJS
verb	base	VB
	non-3rd, present tense	VBP
	3rd person, present	VBZ
	past tense	VBD
	past participle	VBN
	gerund	VBG

Table B.1: Some part-of-speech labels.

```
(letter+){consonant}''ies''          →  (letter+){consonant}''y''+s
(letter+){consonant}''ettes''        →  (letter+){consonant}''ette''+s
(letter+){consonant}{consonant}''es''  →  (letter+){consonant}{consonant}+s
(letter+){vowel}{vowel}''es''        →  (letter+){vowel}{vowel}+s
```

Figure B.1: Rules for stemming nouns

For each word, the program first looks it up in the list of exceptions. If it is there, it returns the stem and the affix. For example, if the input is the word geese, the output is goose+s, where the affix s represent the plural number.

If the word was not in the list, then the stemming rules are used. Figure B.1 shows some examples of rules for stemming nouns. The first rule, for example, means that if the word consists of a sequence of letters followed by a consonant and the string "ies", then the stem is the word up to the last consonant followed by a "y", and the affix is "s".

B.1.4 Chunk parsers

The stemmed text is next passed through three different chunk parsers. They are all built as Transformation List learners [Ramshaw and Marcus, 1995] that learn rules to bracket the text, using as features the part-of-speech and the lexical form of the words in a context around each word [Manandhar and Alfonseca, 2000, Alfonseca, 2000].

The first one learns **complex quantifiers** such as `at least 100`, `hundreds of`, `between 3 and 5`, etc. This chunker was found to be useful if used before the Noun Phrase chunker, because these quantifiers are easy to recognise (they are fixed expressions, with few variations), and they can be easily learnt.

For training the chunker, the parsed version of the Wall Street Journal was used, because some of these complex quantifiers are marked in the parse trees with the label `QP` (Quantifier Phrase). Every sentence containing a constituent of this type was extracted, and the training corpus was constructed as a list of tuples <word, PoS, `IOB`>. PoS represent the part-of-speech of the word, and `IOB` is one of the following labels, `I`, `O`, and `B`, each having the following information [Ramshaw and Marcus, 1995]:

`I`, if the word is inside a base QP;

Precondition		
Word	**part-of-speech**	**IOB tag**
Second word before it	*Any*	*Any*
Previous word	*Any*	O
Current word	DT	I
Next word	*Any*	*Any*
Second word after it	*Any*	*Any*
Action		
take the determiner outside the Complex Quantifier.		

Table B.2: The first rule in the Transformation List for labelling base Quantifier Phrases.

O, if a word is outside;
B, if a word is at the beginning of an QP and the previous word is inside other QP.

In this way, the bracketing problem is changed into a classification problem, in which words in a text have to be labelled with each of the three labels. As an example, Table B.2 contains the first rule in the transformation list, which states that if a determiner is tagged as I (inside a complex quantifier), and the previous word is tagged as O (outside), then the determiner should be retagged as O.

Not every QP was tagged in the parsed files in the Wall Street Journal. Therefore, after the first Transformation List had been generated, it was used to retag the training corpus, and the resulting bracketing was revised by hand. This revision resulted in many more QPs bracketed in the training corpus, and this one was used for obtaining the final Transformation List. The performance of the QP chunker on a test set extracted from other section from the Penn Treebank is roughly 95%.

Secondly, the **Noun Phrase chunker** was trained using the chunked section of the Penn Treebank as training corpus. The accuracy obtained with it is the same that Ramshaw and Marcus [1995] report, around 92%, but we found that many of the errors committed were due to random errors in both in the training and the test corpus. Table B.3 shows the first rule of the generated Transformation List. It says that, if a singular common noun is at the beginning of an NP, and the previous word is inside an NP, then both NPs must be joined in one.

One person-month was spent performing a semi-automatic engineering work, classifying these errors in the training and test corpora, and correcting them by hand, expecting that the learner would find it easier to learn from a clean corpus. Most of the errors were due to the fact that different labelling criteria had been followed by the labellers of the Penn Treebank. The errors found can be classified in the following groups:

- Words which can function in more than one possible ways, such that their labels had been tagged inconsistently throughout the corpus. Examples of this kind of errors are the pairs of distributive conjunctions *either-or* and *neither-not*; in some cases, the part-of-speech of *either* is tagged as a determiner (hence it is mistaken with the determiner *either* in 'he tasted both dishes and didn't like either'); apart from that, these conjunctions were 28 times tagged as O and 5 times tagged as I. Thus, they were automatically pointed out as words that appeared with different labels. After a manual study of the problem, the following conclusions were drawn:
  - If the conjunctions are distributing adjective phrases that complement a noun, then they are inside the base Noun Phrase, e.g.

Precondition		:
Word	**part-of-speech**	**IOB tag**
Second word before it	*Any*	*Any*
Previous word	*Any*	I
Current word	NN	B
Next word	*Any*	*Any*
Second word after it	*Any*	*Any*
Action		
join the noun with the previous word.		

Table B.3: The first rule in the Transformation List for labelling base Noun Phrases.

(17) ... [he$_{PRP}$] does$_{VBZ}$ n't$_{RB}$ expect$_{VB}$ [a$_{DT}$ loss$_{NN}$] in$_{IN}$ [either$_{DT}$ the$_{DT}$ third$_{JJ}$ or$_{CC}$ fourth$_{JJ}$ quarter$_{NN}$] ...

– If the conjunctions are distributing nouns phrases, then they are outside any of the base noun phrases.

(18) [He$_{PRP}$] favors$_{VBZ}$ either$_{CC}$ [an$_{DT}$ all-stock$_{JJ}$ fund$_{NN}$] or$_{CC}$ [a$_{DT}$ balanced$_{JJ}$ fund$_{NN}$] ...

A total of 132 labelling errors were found with this procedure.

- **Words with an unlikely tag:** Some parts-of-speech nearly always appear inside basic noun phrases, while others tend not to. For instance, nouns are usually part of base NPs, and prepositions aren't. Using this criterion, the corpus was reviewed by hand looking for mistagged tokens. Sometimes, a PoS tagging error was detected; other times, it was an IOB labelling error. A total of 5667 labels were corrected at this step.

- **Active learning:** Finally, we programmed the NP chunker to learn to chunk Noun Phrases from the training corpus. Every time it found a case that was difficult to learn, i.e., when the rule generated for the Transformation List to correct the IOB label of the word produced no overall improvement in the whole corpus, then the word and its sentential context appears in the screen, and a human annotator decided whether the PoS and the IOB tags were correct for that word. A total of 343 errors were corrected at this step.

By a combination of the correction of the corpus, and the addition of the QP chunker (which also reduced the complexity of the corpus), this final chunker was able to improve the F-measure up to 94.51%, using the TnT tagger on the test corpus. Finally, the noun chunks are analysed in order to annotate features such as gender, number and person.

Thirdly, yet another Transformation List was learnt, to bracket complex **Verb Phases**, such as sequences of auxiliaries and verbs, or verbs with adverbs inside (e.g *to accurately obtain*). The training corpus was also obtained from the tagged version of the Penn Treebank. After bracketing, the Verb Phrases are analysed and annotated with information about tense, person, number, voice (active or passive), and other information (e.g. whether it contains a modal verb).

B.1.5 Quotes solver

Texts that have been transferred into electronic format using Optical Character Recognition methods are very likely to contain a few degree of errors, specially when distinguishing similar symbols such as the

lowercase 'l' and the number '1'. Not surprisingly, many of those errors happen when trying to recognise the punctuation symbols, such as commas, dots and quotes. This provokes that it is not infrequent to find, in corpora such as LOTR, opening quotes without the corresponding closing quote, and vice versa.

The quotes solver is a module that studies the function that every quote is performing. We have identified the following possible functions:

- Marking a contraction (e.g. `I've`, `haven't` or `D'Ambrosio`)
- Being part of the genitive case marker (e.g. `John's` or `students'`).
- Defining the boundaries of part of a text with some characteristics: either a parenthetical text (e.g. `John ''Killer'' Aton`), or the complement of a verb of communication (e.g. *say*) when a text is cited word by word.

Distinguishing between these few cases is easy, given that they always appear in different contexts. Most contractions are actually identified by the tokeniser, which expands them into the uncontracted form, so there is no need to worry about them; and the genitive case when a quote is followed by a single 's is also simple to identify. However, it requires some analysis of the context to distinguish between the genitive marker after a plural noun, and a closing quote; specially if we know that there will be missing quotes in the text, and therefore an unbalanced quote is not necessarily a genitive case marker.

The procedure followed is the following: firstly, the quotes are labelled as *opening*, *closing* or *ambiguous*. Opening quotes are quotes of the form ' or ' which are preceded by a whitespace and followed by other symbol; and closing quotes are quotes followed by a whitespace and preceded by other symbol. Any other quote is ambiguous (it can be either opening or closing).

Next, quotes are analysed like parentheses: opening quotes are introduced into a stack, and when a closing quote is found, all the text between the quote at the top of the stack and the closing quote is grouped. When the program finishes analysing a paragraph, if the stack is empty, then the paragraph is forgotten and the program skips to the next one. If the stack still has opening quotes, then the closing quote is looked for in the following paragraphs; on the other hand, when there is excess of closing quotes, the paragraph is re-analysed to check whether some of them might be really genitive case markers instead.

B.1.6 Parsing

Finally, several hand-crafted rules have been written for identifying subject-verb and verb-object relationships, as well as some prepositional phrase attachment in cases which are not ambiguous. This shallow parser is able to identify the subject-verb and the verb-object relations for around one half of the verbs in the texts analysed.

B.2 Changes to the WordNet structure

WordNet, as a general purpose semantic lexicon, has resulted extremely useful for this work. However, there were some characteristics of WordNet that have been changed in order to adapt it to the requirements of the architecture. Only those changes that were absolutely necessary, and which could be implemented without altering its internal structure were done. In this way, it was intended to facilitate as much as possible the porting of these changes when a new version of WordNet appears.

The changes done are the following:

1. A mechanism for handling microtheories.

2. Distinction between concepts and instances.

The following sections describe how they have been implemented.

B.2.1 Microtheories

WordNet was aimed to cover general-purpose concepts in the English language. However, when we treat documents from any specific area, we need to know the jargon, i.e., the words and concepts particular to that field of knowledge. The off-line processing performed by the system on the source texts tries to enrich WordNet with new domain-specific terms. However, adding these concepts directly to WordNet would provoke several unwanted problems. Firstly, as Knowledge Bases get increasingly complex with more and more information, pertaining to different domains, maintenance of the concept network usually becomes very hard or even impossible. Secondly, if a semantic network includes concepts from many different areas at the same time, then probably many of them will be irrelevant for most applications. For example, a dictionary entry for **man** referring to the Unix *manual* command is probably irrelevant for most applications, and extra work for Word Sense Disambiguation will have to be performed. Finally, if we simply add the new synsets into WordNet, then we lose compatibility with the standard versions.

An easy solution is the subdivision of the whole network in smaller portions or microtheories. In this way, the domain-specific lexicon is kept apart from the general-purpose words, and loaded only when necessary. This solution has been applied to large Knowledge Bases in projects such as Cyc [Lenat, 1995], OntoLingua [Farquhar et al., 1997], TOVE [Fox and Gruninger, 1994] and the work described by Clark and Porter [1997].

In the new implementation, WordNet has been sub-divided into a multinetwork of microtheories. The basic WordNet 1.7 is considered as the kernel microtheory, and any new addition extracted from texts is classified in a different microtheory according to its domain. This has the advantage that the system can store knowledge about:

- Different domains, so that the assertions in one domain are chosen more likely than assertions in different domains. For example, in biology an *ant* is an animal, whilst in Computer Science an *ant* is an *agent* used for experimenting in simulation of biological systems. If the user introduces an assertion or a query concerning an *ant*, and we can infer the domain of expertise from the context, we can retrieve directly the information relevant to the topic.

- Different users. If this Lexical Knowledge Base is used in a program that adapts to different users, it can store their particular characteristics and interests so it can answer the information that is relevant to him or her. For instance, users can have separate microtheories with statements about their preferences, or about the concepts that they know.

To implement microtheories in WordNet, the design allows pointers to classes in other microtheories. For example, *hobbit* is a hyponym of *person*, which appears in the WordNet kernel network. If we introduce *hobbit* in a microtheory about *The Lord of the Rings*, the hyperonym link from *hobbit* to *person* crosses a microtheory boundary. And it will be considered only when both microtheories involved are of interest to the application.

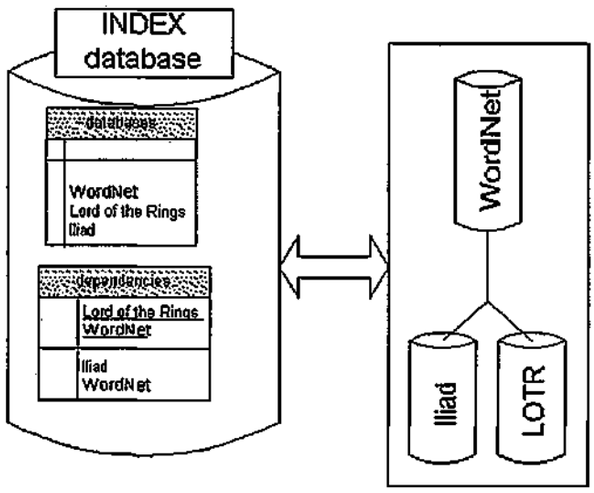


Figure B.2: Structure of the general WordNet database with two microtheories: one extracted from *The Lord of the Rings* and the other extracted from *The Iliad*. Both depend on the basic WordNet taxonomy, which means that both have links to other synsets in it.

High level design of the database

The implementation uses the `mysql` database, that is portable across several Operating Systems, and the following databases have been included:

- A database called `wndb_index` that keeps track of all the microtheories that have been generated since the installation of the system, and the dependences among them.
- A database called `WordNet`, that contains the synsets and relations from the original WordNet 1.7 semantic network.
- A different database for each domain studied.

For example, the two databases `lotr` and `iliad` were generated while studying *The Lord of the Rings* [Tolkien, 1968] and *The Iliad* [Homer], respectively. Each one of them depends on the `WordNet` database, which means that its links may refer to WordNet synsets. Therefore, if any of them is selected to be used, WordNet is automatically selected as well. This situation is displayed in figure B.2.

B.2.2 Instances and concepts

As already mentioned before, the Term Classification procedure required to know which of the unknown terms are concepts and which are instances, for calculating the initial order in which the concepts will be learnt. In Section 5.4.1, *Partition of instances and concepts*, there is a description about how the new terms are classified as instances and concepts, using a Maximum Entropy approach. This section describes the design decisions about what will be considered an *instance*, and how the manual annotation of WordNet was performed.

Many taxonomies handle two different kinds of objects. A **concept** represents a set of things of interest that have something in common; while an **instance** is a single example of a concept. For illustration, *human*

is a concept that can denote each one of the instances from the *Homo sapiens* species; while *Shakespeare* is an instance of that concept, and denotes a particular human. This distinction will be made clear in the following sections. Usually, but not always, common nouns represent concepts and proper nouns represent instances.

As far as I know, there has been little attention to the fact that WordNet contains both concepts and instances in the semantic network of nouns, with no distinction between them. However, it would be useful for many applications to know which synsets are concepts and which ones are instances, because they have different properties. Other taxonomic resources, such as Cyc [Lenat and Guha, 1990] and Ontolingua [Farquhar et al., 1997], have implemented this distinction.

Motivation

Apart from making WordNet more similar to other existing ontologies, such as Cyc or ontologies developed using Knowledge Representation Systems like Ontolingua, thence facilitating the interchange of information between these systems, this work was necessary for the Text Classification algorithm. If instances are marked as such, then they are automatically non-candidates to be hyperonyms of a new unknown word, because only concepts can be hyperonyms. For example, if a program is analysing texts about sailing and it finds the word *pram* referring to a sailboat, it may suggest the WordNet synset *sailboat* as a possible hyperonym, but it will never suggest a synset such as *Mayflower*, because it refers to an instance of a boat, and cannot have hyponyms.

Other application particular to this algorithm is for reordering domain-dependent concepts before introducing them inside the semantic network. Let us suppose that a text introduces a new concept, such as *hobbit*, and describes instances of that new concept (e.g. *Frodo*, *Bilbo*, etc.) In this case, it will be useful to extend WordNet first with the unknown concepts, and next with the instances. Therefore, when *Frodo* is added to the ontology, *hobbit* has already been placed there and can be considered as a candidate hyperonym.

A taxonomy with instances and concepts

Let us define the semantic network of nouns in WordNet as a tuple $\mathcal{W} = (\mathcal{L}, \mathcal{S}, f_{\mathcal{L}}, h_{\mathcal{S}}, \mathcal{R})$ where

- $\mathcal{L}$ is the set of lexical entries (words).
- $\mathcal{S}$ is the set of synsets.
- $f_{\mathcal{L}} : \mathcal{L} \rightarrow \mathcal{S}^*$ is a function that links the lexical entries with the synsets that contain them.
- $h_{\mathcal{S}} : \mathcal{S} \rightarrow \mathcal{S}^*$, called *hyperonymy*, arranges the concepts and instances in a hierarchy.
- $\mathcal{R}$ is the set of other relationships.

Inside $\mathcal{S}$, there are two different kinds of entities: concepts and instances, as defined by Degen et al. [2001] (he calls instances *individuals*, and concepts *universals*):

> Individuals belong to the realm of concrete entities, which means that they exist within the confines of space and time. Universals, in contrast, are entities that can be instantiated simultaneously by a multiplicity of different individuals that are similar in given respects. We can think of universals as patterns of features which are realized by their instances.

There is disagreement about the interpretation of instances and concepts. One of the most widely accepted considers concepts as the sets of their instances [Montague, 1974]. Using an example from [Welty

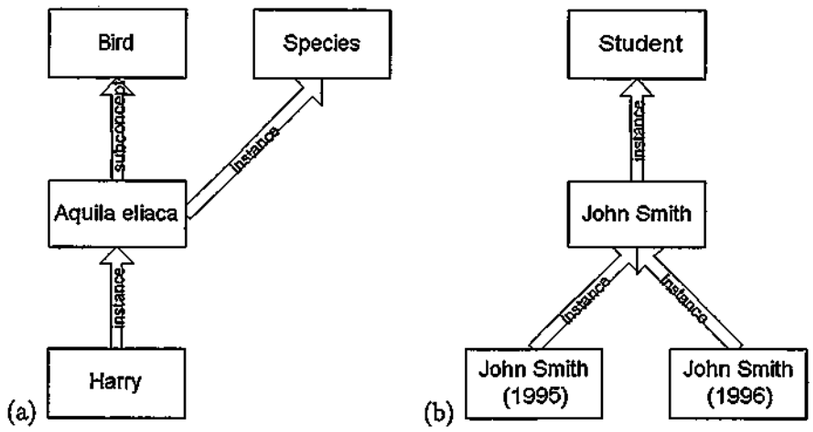


Figure B.3: (a) In this taxonomy, *Aquila eliaca* is at the same time a concept and an instance [Welty and Ferucci, 1999]. (b) Although the student *John Smith* would normally be used as an instance, in some occasions it may be useful to consider it as a concept.

and Ferucci, 1999], *bird* would be the set containing all possible birds, real and imagined, past and future; *eagle* would be a subset of *bird*, and *the eagle Harry* would be an instance (a member) of both sets.

On the other hand, there are theories that consider instances as sets of universals, or sets of individualised properties, also called tropes. However, as Degen et al. [2001] points out, this interpretation poses some difficulties in dealing with different temporal profiles of the entities.

Instances, or instances of something else

According to the definition stated above, when a synset represents features that can be realised by distinguishable instances, it is considered a concept, while when it refers to a concrete entity, or different manifestations of it, then it is an instance. But this definition still has to be further clarified.

As Welty and Ferucci [1999] notes, something can be both a concept and an instance. He proposes an example in which we have four synsets: *species*, *bird*, *Imperial eagle (Aquila eliaca)*, and *Harry*, which is an imperial eagle. Here *Aquila eliaca* is an instance of *species*, but at the same time it is a concept, one of whose instances is *Harry* (see Figure B.3a).

In fact, everything can behave as an instance or a concept, depending on the interpretation. For example, *the McMillans* can be an an instance of *family* or *clan*, but it is as well a concept that includes all its members, as in sentence (19).

(19) That man is a McMillan

Or a University software might be interested in creating new entries every year for each student. In that context, *John Smith-1995* and *John Smith-1996* can be considered instances of *John Smith*, which is an instance of *student* (see Figure B.3b). Summarising,

- On the first hand, every concept is an instance of *Concept*.
- On the second hand, every instance may be considered a concept whose instances are different manifestations of that same concept (e.g. at different times or observed by different people.)

The previous reasoning indicates that, rather than classifying synsets as instances or concepts, we should label the hyperonymy links as instance links or sub-concept links instead. However, by doing this we would lose the intended objective of making WordNet more similar to other existing lexical databases or Knowledge Representation Systems, such as Cyc or Ontolingua, in which entities have to be either instances or concepts. Therefore, the following middle position has been adopted for this work:

- Synsets are classified either as concepts or instances.
- If a synset has hyponyms, it is being considered as a concept in the taxonomy, and thence it is marked it as such.
- Leaf synsets (synsets with no hyponyms) will be annotated either as *instances* or *concepts*, according to the relation they hold with respect to to their immediate hyperonyms in the taxonomy.
- If a leaf synset has several immediate hyperonyms, and it is a subconcept of at least one of them, then it will be classified as a concept, because there is at least one interpretation in which it will have its own instances.

As discussed, probably this is not the optimal solution according to the theory, but it was considered a good criterion in order to make WordNet more similar to other existing taxonomic resources.

Changes to WordNet

If $\mathcal{S}$ is the set of synsets in WordNet, and $h_{\mathcal{S}}$ is the hyperonymy relationship, $\mathcal{S}$ will be divided into two subsets $\mathcal{C}$ and $\mathcal{I}$, and $h_{\mathcal{S}}$ will be modified in the following way:

- $\mathcal{C}$ will be a set of concepts.
- $\mathcal{I}$ will be a set of instances.
- $\mathcal{S} = \mathcal{C} \cup \mathcal{I}$
- $h_{\mathcal{S}}$ is modified to $h_{\mathcal{S}} : \mathcal{S} \rightarrow \mathcal{C}^*$.

Hence the definition of $\mathcal{W}$ is modified to include the instances: $\mathcal{W} = (\mathcal{L}, \mathcal{S}, \mathcal{I}, f_{\mathcal{L}}, h_{\mathcal{S}}, \mathcal{R})$. If we define a leaf as any synset that has no hyponyms,

$$Leaves(\mathcal{W}) = \{s \epsilon \mathcal{S}, \nexists n : h_{\mathcal{S}}(n) = s\}$$

then, instances can only be leaves in the WordNet taxonomy. However, some leaves can represent concepts, if they have not been instantiated: $\mathcal{I} \subseteq Leaves(\mathcal{W})$.

Manual annotation of WordNet

In English, the instances of some concepts are rarely named. These concepts include *psychological features*, *acts*, etc. For example *the fear I felt yesterday at 12 noon* is an instance of the concept *fear*. Although it is possible to give it a name such as *My Midday Fear* or any other identifier, these kinds of entities do not usually receive a proper name in the English language. In English, concepts whose instances are usually named are *animate beings* (e.g. people, animals, even plants), *locations* (e.g. cities, etc.), *ideas and intellectual works* (e.g. theorems, books, etc.) and some objects (e.g. ships, such as *Mayflower*).

However, after examining WordNet in detail, the conclusion was that the language can contain, in theory, instances of practically every concept. A few examples of instances of unlikely entities are:

synset id	synset word	concept-leaves	instance-leaves
n00005145	person	4,534	2,913
n00018241	location	735	1,773
n00016210	psychological_feature	2,558	618
n00015211	artifact	7,494	120
n00021056	act, human_action	4,213	210
other		32,019	1,399
Total		51,553	7,033

Table B.4: Results of the manual annotation of instances and concepts in WordNet (only the 51,553 leaves are considered; the rest of the synsets are all considered concepts).

- *Creation*, meaning *God's act of bringing the universe into existence*, which is a hyponym of *action*.
- *Gettisburg's Address*, which is a speech addressed by Abraham Lincoln during the war, and is a hyponym of *speech_act*.

Therefore, all leaf synsets in the nouns taxonomy have been manually classified, according to the relation they hold with their hyperonyms.

Manual annotation results

WordNet version 1.7 contains 58,586 leaf-synsets, i.e. synsets with no hyponyms. All of them were manually annotated according to the criteria described above, and the results are displayed in Table B.4. If the annotations are correct, there are 51,553 concepts and 7,033 instances among the leaves. Some of the branches with a high number of instances are *person*, *location* and *psychological feature*, this last branch because it includes all the mythological characters.

Annotating difficult cases

Language is always changing, and something that is considered an instance at a certain moment can, with time, come to be a concept. For example, the first *Unix* could be considered, at the moment it was released, an instance of the concept *operating system*. However, it can now be considered as any of the operating systems that have a similar architecture and a common set of commands, and that includes, among others, *solaris*, *BSD Unix*, *AIX*, *IRIX* and *Linux*. As said above, when there exists a plausible interpretation in which a synset can be considered a concept, it has been classified as such.

Other decisions were also difficult to take because the meaning of the synset had different points of view. For example, literary works such as *Genesis* or *Aesop's Fables*, can be interpreted, depending on the context, in different ways, as the following sentences show. In (20a), *Genesis* refers to the text contents, the intellectual work (an abstraction); and in (20b) it refers to a set of pages in a book. *Aesop's Fables* in (20c) also refer to the physical book. A theory about the same word representing different views of the same thing was developed mainly by Pustejovsky [1995].

(20) a. Genesis was translated to Greek.

b. The boy tore off Genesis from his Bible.

c. Aesop's Fables looks nice on the shelf.

avatar
=> Jagannath
=> Kalki
=> Krishna
=> Rama
=> Ramachandra
=> Balarama
=> Parashurama

Figure B.4: *Rama* should be an instance of *avatar*, but it is also a concept which has three different instances: the three incarnations. (Repeated from Figure 5.11).

In WordNet both *Genesis* and *Aesop's Fables* are hyponyms of *abstraction*, but not of *object*. Therefore, the synsets in WordNet refer to the meaning in (20a), not the remaining ones. As such, they refer to the textual contents of the books, not any physical print, and they were classified as instances.

Some concepts can have different names depending on their manifestations. For example, the planet *Venus* can be called *morning star* if it is visible in the early morning, or *evening star* if visible at sunset. If we were annotating the hyperonymy relationships, it would be possible to set that *Venus* is an instance of *planet*, and both *morning star* and *evening star* are instances of *Venus*. However, as only synsets have been annotated, instead of relations, a synset cannot be a concept and an instance at the same time. A similar case is shown in Figure B.4.

Automatic annotation of WordNet

After the manual annotation, a machine learning algorithm has been trained in order to be able to classify future synsets as instances or concepts. There were two motivations for this work. In the first place, the automatic procedure can be used to predict whether a new domain-specific concept, not present in WordNet, is an instance or a concept without the need of human annotators. Secondly, by using very simple features, it can be shown that most instances and concepts, in language, are indeed used in different ways and can be easily detected, i.e. there is some empirical evidence that it is a difference that really exists in language.

The learning model chosen is a Maximum Entropy model [Berger et al., 1996] [Ratnaparkhi, 1998]. In this framework, the problem consists in learning a probability model

$$P_{ME}(s) = \frac{1}{Z} \cdot P_0(s) \cdot \exp\left(\sum_i \lambda_i f_i(s)\right)$$

where $P_{ME}(s)$ is the probability that s is an instance, $P_0(s)$ is an initial probability distribution, Z is a normalising constant, and f_i are binary features about the examples. Using an iterative algorithm, it is possible to obtain values for the parameters λ_i so that the model classifies the training data as best as possible. The implementation uses the Java package `quipu.maxent`, which is freely distributed [Baldridge et al., 2001].

Features chosen

Instances, in language, have some properties in common with mass nouns. For example, they are rarely preceded by articles *the* and *a* or used in plural number. On the other hand, mass nouns can be quantified with weight, volume, etc. while instances cannot. These facts were used for choosing the right features to distinguish them.

At this point, it is necessary to make a distinction. Some instance names, such as *Judas* —denoting the man that lived in Judea (sense 08885770)— have undertaken other meanings such as *someone who betrays* (sense 08201644). These are different synsets: the first one represents an instance and, as such, cannot have an article in the specifier position; and the second one represents a concept and can be quantified or preceded by a determiner.

(21) a. Judas hung himself.

b. Don't trust him, I think he's a Judas.

c. There are several 'Judases' in that political party.

The following are several examples of features:

$f_1(s) = true$ if any word in synset s was found preceded by the determiner *the* in the documents; *false* otherwise

$f_2(s) = true$ if no word in synset s was found with any determiner in the sample documents; *false* otherwise

Capitalisation was also used as a feature. Although not every capitalised word is an instance, many of them are, so this feature provides support in favour of considering the word an instance.

$f_3(s) = true$ if any every word in synset s is capitalised; *false* otherwise

Internet has been used as a corpus to collect features about the way words are used. The procedure for collecting documents containing the words from some WordNet synset has been already described in Section 5.5.1.

Results

The training set was built with 300 concepts and 150 instances selected randomly from among the WordNet leaves. The classifier was tested with a five-fold evaluation: the training set was divided in five subsets of the same size, each one with 60 concepts and 30 instances; in each experiment, four of them were chosen for training and the fifth for testing. The program automatically downloaded the documents for each synset and extracted the features. To measure the accuracy of this procedure, the manual annotation of WordNet was used as the set of correct labels. Finally, a baseline was calculated considering that every synset is a concept. This has an accuracy of $\frac{51,553}{58,586} = 88\%$.

The resulting accuracy for each of the five experiments is shown in Table B.5. The final accuracy is 96.62%, with a mean of three mistakes in each test set. This result indicates that the features chosen are useful for the classification, a fact which has the implication that instances are indeed used in a different way in language, and they can be recognised using just a very small set of features.

experiment	accuracy
1st.	95.6%
2nd.	98.9%
3rd.	94.3%
4th.	98.9%
5th.	95.4%
Mean	96.62%
Baseline	88%

Table B.5: Results of the five-fold evaluation

Analysis of the errors

By examining the erroneous classifications by hand, it was noted that the major source of errors were synsets that describe languages, such as *English*, *French*, *Chinese*, etc. They had been manually annotated as *concepts*, because a language can be considered as the concept that groups all its dialects. This decision was based, as well, in the fact the the synset for the *English language* (synset number 05689601) has as many as eight hyponyms, which refer to the more common English dialects: *American English*, *cockney*, *Middle English*, etc.

However, languages are usually used without determiners, never in plural, and written capitalised, so the automatic algorithm misclassified them. The addition of the new feature

$f_4(s) = true$ if it is defined as a *language* or a *dialect* in the synset gloss; *false* otherwise

increased the accuracy to 97.2%.

The remaining classification errors were due to a variety of reasons: *1530s* and *1770s* were considered as plural words, and were misclassified as concepts; or *Allen_wrench* never appeared in the sample documents in plural, so it was finally mistagged as an instance. In fact, the other ten misclassifications were due to a lack of enough data, because words that can be used with determiners or in plural form never appeared like that in the small corpora downloaded from Internet.

Section 5.4.1 describes how this algorithm behaved when applied on newly learnt terms, such as unknown words from *The Lord of the Rings* and *The Iliad*.

Conclusions

The work described here aimed at identifying concepts and instances in WordNet. This kind of information, present in other lexical knowledge bases, such as Cyc [Lenat and Guha, 1990], was useful later for the Text Classification component.

As discussed in section B.2.2, it is probably more formal to annotate the hyperonymy relationships as *instance-of* and *subclass* relations, given that the same synset can be considered an instance or a class depending on the interpretation. Nevertheless, for several reasons, it was decided that it was better, in this case, to annotate the synsets. In this way, compatibility with other popular LKBs can be more easily guaranteed. The annotation of the relationships is an open work that might be attempted in the future.

Our experimental results show that with a very reduced set of features (capitalisation and determiners), and a rather small training set, a high accuracy can be obtained in distinguishing instances and concepts.

Appendix C

Example of generated summaries

This appendix contains some examples of pages that were generated by the system.

The first few examples contain summaries generated, from the same original textual data, for users that connect with different profiles. The original text is the first section in the first chapter of *The Voyages of the Beagle*. When a new user enters the system and starts to read the text sequentially, that is the first page that is generated for him or her.

Let us suppose that a user selects only one pre-defined interest. There are three distinct stereotypes defined for Darwin's text: history (which selects narrative text, describing the events that happened to Darwin: voyages, trips, etc.), biology (which selects the portions of the text that refer to life forms), and geography (which selects the portions of the text that contain description of places and peoples). Figure C.1 shows the text adapted to the user interested in narrative text; Figures C.2 and C.3 contain the section for a user interested in biology; and Figures C.4 and C.5 show the section for a user interested in geography.

Let us suppose now that another user arrives who is interested in all of Darwin's text. In this case, he or she will select the boxes for the three stereotypes: biology, history and geography, so every paragraph will be selected for presentation. However, it may be the case that the generated site, under this condition, will be too large for that user. A possible way to reduce it is to tell the system to summarise the output. Figures C.6 and C.7 show the 30% summary of the first section of the first chapter *The Voyages of the Beagle*, as shown to this user.

Finally, let us suppose that any of the mentioned users clicks some hyperlinks that opens the page with information automatically collected from the Internet about the concepts mentioned in the hypertext pages. Figures C.8 and C.9 show the pages collected about the concepts *Rio* (Rio de Janeiro, in Brazil) and *Cornwall*.

Chapter 1. St. Jago– Cape de Verd Islands

After having been twice driven back by heavy south-western gales, Her Majesty's ship "Beagle," a ten-gun brig, under the command of Captain Fitz Roy, R.N., sailed from Devonport on the 27th of December, 1831. The object of the expedition was to complete the survey of Patagonia and Tierra del Fuego, commenced under Captain King in 1826 to 1830 –to survey the shores of Chile, Peru, and of some islands in the Pacific– and to carry a chain of chronometrical measurements round the World. On the 6th of January we reached Teneriffe, but were prevented landing, by fears of our bringing the cholera: the next morning we saw the sun rise behind the rugged outline of the Grand Canary Island, and suddenly illumine the Peak of Teneriffe, whilst the lower parts were veiled in fleecy clouds. This was the first of many delightful days never to be forgotten. On the 16th of January 1832 we anchored at Porto Praya, in St. Jago, the chief island of the Cape de Verd archipelago.

One day, two of the officers and myself rode to Ribeira Grande, a village a few miles eastward of Porto Praya. Until we reached the valley of St. Martin, the country presented its usual dull brown appearance; but here, a very small rill of water produces a most refreshing margin of luxuriant vegetation. In the course of an hour we arrived at Ribeira Grande, and were surprised at the sight of a large ruined fort and cathedral. This little town, before its harbour was filled up, was the principal place in the island: it now presents a melancholy, but very picturesque appearance. Having procured a black Padre for a guide, and a Spaniard who had served in the Peninsular war as an interpreter, we visited a collection of buildings, of which an ancient church formed the principal part. It is here the governors and captain-generals of the islands have been buried. Some of the tombstones recorded dates of the sixteenth century. (1/2. The Cape de Verd Islands were discovered in 1449 . There was a tombstone of a bishop with the date of 1571; and a crest of a hand and dagger, dated 1497.) The heraldic ornaments were the only things in this retired place that reminded us of Europe. The church or chapel formed one side of a quadrangle, in the middle of which a large clump of bananas were growing. On another side was a hospital, containing about a dozen miserable-looking inmates.

We returned to the Venda to eat our dinners. A considerable number of men, women, and children, all as black as jet, collected to watch us. Our companions were extremely merry; and everything we said or did was followed by their hearty laughter. Before leaving the town we visited the cathedral. It does not appear so rich as the smaller church, but boasts of a little organ, which sent forth singularly inharmonious cries. We presented the black priest with a few shillings, and the Spaniard, patting him on the head, said, with much candour, he thought his colour made no great difference. We then returned, as fast as the ponies would go, to Porto Praya.

Figure C.1: Contents of the first page shown to a user that starts to browse the adaptive site about *The Voyages of the Beagle* in sequential order, for a profile on **history** (narrative text).

Chapter 1. St. Jago– Cape de Verd Islands

Generally the atmosphere is hazy; and this is caused by the falling of impalpably fine dust, which was found to have slightly injured the astronomical instruments. The morning before we anchored at Porto Praya, I collected a little packet of this brown-coloured fine dust, which appeared to have been filtered from the wind by the gauze of the vane at the masthead. Mr. Lyell* has also given me four packets of dust which fell on a vessel a few hundred miles northward of these islands. Professor Ehrenberg* finds that this dust consists in great part of infusoria with siliceous shields, and of the siliceous tissue of plants. (1/3. I must take this opportunity of acknowledging the great kindness with which this illustrious naturalist has examined many of my specimens. I have sent (June 1845) a full account of the falling of this dust to the Geological Society.) In five little packets which I sent him, he has ascertained no less than sixty-seven different organic forms!. The infusoria, with the exception of two marine species, are all inhabitants of fresh-water. I have found no less than fifteen different accounts of dust having fallen on vessels when far out in the Atlantic. From the direction of the wind whenever it has fallen, and from its having always fallen during those months when the harmattan is known to raise clouds of dust high into the atmosphere, we may feel sure that it all comes from Africa. It is, however, a very singular fact, that, although Professor Ehrenberg* knows many species of infusoria peculiar to Africa, he finds none of these in the dust which I sent him. On the other hand, he finds in it two species which hitherto he knows as living only in South America. The dust falls in such quantities as to dirty everything on board, and to hurt people's eyes; vessels even have run on shore owing to the obscurity of the atmosphere. It has often fallen on ships when several hundred, and even more than a thousand miles from the coast of Africa, and at points sixteen hundred miles distant in a north and south direction. In some dust which was collected on a vessel three hundred miles from the land, I was much surprised to find particles of stone above the thousandth of an inch square, mixed with finer matter. After this fact one need not be surprised at the diffusion of the far lighter and smaller sporules of cryptogamic plants.

During our stay, I observed the habits of some marine animals. A large Aplysia is very common. This sea-slug is about five inches long; and is of a dirty yellowish colour, veined with purple. On each side of the lower surface, or foot, there is a broad membrane, which appears sometimes to act as a ventilator, in causing a current of water to flow over the dorsal branchiae or lungs. It feeds on the delicate seaweeds which grow among the stones in muddy and shallow water; and I found in its stomach several small pebbles, as in the gizzard of a bird. This slug, when disturbed, emits a very fine purplish-red fluid, which stains the water for the space of a foot around. Besides this means of defence, an acrid secretion, which is spread over its body, causes a sharp, stinging sensation, similar to that produced by the Physalia, or Portuguese man-of-war.

Figure C.2: Contents of the first page shown to a user that starts to browse the adaptive site about *The Voyages of the Beagle* in sequential order, for a profile on **biology** (continued in next figure).

I was much interested, on several occasions, by watching the habits of an Octopus, or cuttle-fish. Although common in the pools of water left by the retiring tide, these animals were not easily caught. By means of their long arms and suckers, they could drag their bodies into very narrow crevices; and when thus fixed, it required great force to remove them. At other times they darted tail first, with the rapidity of an arrow, from one side of the pool to the other, at the same instant discolouring the water with a dark chestnut-brown ink. These animals also escape detection by a very extraordinary, chameleon-like power of changing their colour. They appear to vary their tints according to the nature of the ground over which they pass: when in deep water, their general shade was brownish purple, but when placed on the land, or in shallow water, this dark tint changed into one of a yellowish green. The colour, examined more carefully, was a French grey, with numerous minute spots of bright yellow: the former of these varied in intensity; the latter entirely disappeared and appeared again by turns. These changes were effected in such a manner that clouds, varying in tint between a hyacinth red and a chestnut-brown, were continually passing over the body. (1/4. So named according to Patrick Symes's nomenclature.) Any part, being subjected to a slight shock of galvanism, became almost black: a similar effect, but in a less degree, was produced by scratching the skin with a needle. These clouds, or blushes as they may be called, are said to be produced by the alternate expansion and contraction of minute vesicles containing variously coloured fluids. (1/5. See "Encyclopedia of Anatomy and Physiology" article "Cephalopoda".).

This cuttle-fish displayed its chameleon-like power both during the act of swimming and whilst remaining stationary at the bottom. I was much amused by the various arts to escape detection used by one individual, which seemed fully aware that I was watching it. Remaining for a time motionless, it would then stealthily advance an inch or two, like a cat after a mouse; sometimes changing its colour: it thus proceeded, till having gained a deeper part, it darted away, leaving a dusky train of ink to hide the hole into which it had crawled.

While looking for marine animals, with my head about two feet above the rocky shore, I was more than once saluted by a jet of water, accompanied by a slight grating noise. At first I could not think what it was, but afterwards I found out that it was this cuttle-fish, which, though concealed in a hole, thus often led me to its discovery. That it possesses the power of ejecting water there is no doubt, and it appeared to me that it could certainly take good aim by directing the tube or siphon on the under side of its body. From the difficulty which these animals have in carrying their heads, they can not crawl with ease when placed on the ground. I observed that one which I kept in the cabin was slightly phosphorescent in the dark.

Figure C.3: Contents of the first page shown to a user that starts to browse the adaptive site about *The Voyages of the Beagle* in sequential order, for a profile on **biology** (continued from the previous figure).

Chapter 1. St. Jago– Cape de Verd Islands

After having been twice driven back by heavy south-western gales, Her Majesty's ship "Beagle," a ten-gun brig, under the command of Captain Fitz Roy, R.N., sailed from Devonport on the 27th of December, 1831. The object of the expedition was to complete the survey of Patagonia and Tierra del Fuego, commenced under Captain King in 1826 to 1830 –to survey the shores of Chile, Peru, and of some islands in the Pacific– and to carry a chain of chronometrical measurements round the World. On the 6th of January we reached Teneriffe, but were prevented landing, by fears of our bringing the cholera: the next morning we saw the sun rise behind the rugged outline of the Grand Canary Island, and suddenly illumine the Peak of Teneriffe, whilst the lower parts were veiled in fleecy clouds. This was the first of many delightful days never to be forgotten. On the 16th of January 1832 we anchored at Porto Praya, in St. Jago, the chief island of the Cape de Verd archipelago.

The neighbourhood of Porto Praya, viewed from the sea, wears a desolate aspect. The volcanic fires of a past age, and the scorching heat of a tropical sun, have in most places rendered the soil unfit for vegetation. The country rises in successive steps of table-land, interspersed with some truncate conical hills, and the horizon is bounded by an irregular chain of more lofty mountains. The scene, as beheld through the hazy atmosphere of this climate, is one of great interest; if, indeed, a person, fresh from sea, and who has just walked, for the first time, in a grove of cocoa-nut trees, can be a judge of anything but his own happiness. The island would generally be considered as very uninteresting, but to any one accustomed only to an English landscape, the novel aspect of an utterly sterile land possesses a grandeur which more vegetation might spoil. A single green leaf can scarcely be discovered over wide tracts of the lava plains; yet flocks of goats, together with a few cows, contrive to exist. It rains very seldom, but during a short portion of the year heavy torrents fall, and immediately afterwards a light vegetation springs out of every crevice. This soon withers; and upon such naturally formed hay the animals live. It had not now rained for an entire year. When the island was discovered, the immediate neighbourhood of Porto Praya was clothed with trees (1/1. I state this on the authority of Dr. E. Dieffenbach, in his German translation of the first edition of this Journal), the reckless destruction of which has caused here, as at St. Helena, and at some of the Canary islands, almost entire sterility. The broad, flat-bottomed valleys, many of which serve during a few days only in the season as watercourses, are clothed with thickets of leafless bushes. Few living creatures inhabit these valleys. The commonest bird is a kingfisher (Dacelo Iagoensis), which tamely sits on the branches of the castor-oil plant, and thence darts on grasshoppers and lizards. It is brightly coloured, but not so beautiful as the European species: in its flight, manners, and place of habitation, which is generally in the driest valley, there is also a wide difference. One day, two of the officers and myself rode to Ribeira Grande, a village a few miles eastward of Porto Praya. Until we reached the valley of St. Martin, the country presented its usual dull brown appearance; but here, a very small rill of water produces a most refreshing margin of luxuriant vegetation. In the course of an hour we arrived at Ribeira Grande, and were surprised at the sight of a large ruined fort and cathedral. This little town, before its harbour was filled up, was the principal place in the island: it now presents a melancholy, but very picturesque appearance. Having procured a black Padre for a guide, and a Spaniard who had served in the Peninsular war as an interpreter, we visited a collection of buildings, of which an ancient church formed the principal part. It is here the governors and captain-generals of the islands have been buried. Some of the tombstones recorded dates of the sixteenth century. (1/2. The Cape de Verd Islands were discovered in 1449. There was a tombstone of a bishop with the date of 1571; and a crest of a hand and dagger, dated 1497.) The heraldic ornaments were the only things in this retired place that reminded us of Europe. The church or chapel formed one side of a quadrangle, in the middle of which a large clump of bananas were growing. On another side was a hospital, containing about a dozen miserable-looking inmates.

Figure C.4: Same page shown for a profile on **geography** and descriptions (cont. in next figure).

Another day we rode to the village of St. Domingo, situated near the centre of the island. On a small plain which we crossed, a few stunted acacias were growing; their tops had been bent by the steady trade-wind, in a singular manner– some of them even at right angles to their trunks. The direction of the branches was exactly north-east by north, and south-west by south, and these natural vanes must indicate the prevailing direction of the force of the trade-wind. The travelling had made so little impression on the barren soil, that we here missed our track, and took that to Fuentes. This we did not find out till we arrived there; and we were afterwards glad of our mistake. Fuentes is a pretty village, with a small stream; and everything appeared to prosper well, excepting, indeed, that which ought to do so most– its inhabitants. The black children, completely naked, and looking very wretched, were carrying bundles of firewood half as big as their own bodies.

The scenery of St. Domingo possesses a beauty totally unexpected, from the prevalent gloomy character of the rest of the island. The village is situated at the bottom of a valley, bounded by lofty and jagged walls of stratified lava. The black rocks afford a most striking contrast with the bright green vegetation, which follows the banks of a little stream of clear water. It happened to be a grand feast-day, and the village was full of people. On our return we overtook a party of about twenty young black girls, dressed in excellent taste; their black skins and snow-white linen being set off by coloured turbans and large shawls. As soon as we approached near, they suddenly all turned round, and covering the path with their shawls, sung with great energy a wild song, beating time with their hands upon their legs. We threw them some vintems, which were received with screams of laughter, and we left them redoubling the noise of their song.

One morning the view was singularly clear; the distant mountains being projected with the sharpest outline, on a heavy bank of dark blue clouds. Judging from the appearance, and from similar cases in England, I supposed that the air was saturated with moisture. The fact, however, turned out quite the contrary. The hygrometer gave a difference of 29.6 degrees, between the temperature of the air, and the point at which dew was precipitated. This difference was nearly double that which I had observed on the previous mornings. This unusual degree of atmospheric dryness was accompanied by continual flashes of lightning. Is it not an uncommon case, thus to find a remarkable degree of aerial transparency with such a state of weather?.

The geology of this island is the most interesting part of its natural history. On entering the harbour, a perfectly horizontal white band in the face of the sea cliff, may be seen running for some miles along the coast, and at the height of about forty-five feet above the water. Upon examination, this white stratum is found to consist of calcareous matter, with numerous shells embedded, most or all of which now exist on the neighbouring coast. It rests on ancient volcanic rocks, and has been covered by a stream of basalt, which must have entered the sea when the white shelly bed was lying at the bottom. It is interesting to trace the changes, produced by the heat of the overlying lava, on the friable mass, which in parts has been converted into a crystalline limestone, and in other parts into a compact spotted stone. Where the lime has been caught up by the scoriaceous fragments of the lower surface of the stream, it is converted into groups of beautifully radiated fibres resembling arragonite. The beds of lava rise in successive gently-sloping plains, towards the interior, whence the deluges of melted stone have originally proceeded. Within historical times no signs of volcanic activity have, I believe, been manifested in any part of St. Jago. Even the form of a crater can but rarely be discovered on the summits of the many red cindery hills; yet the more recent streams can be distinguished on the coast, forming lines of cliffs of less height, but stretching out in advance of those belonging to an older series: the height of the cliffs thus affording a rude measure of the age of the streams.

Figure C.5: Same page for a profile on **geography** (cont. from the prev. figure).

Chapter 1. St. Jago– Cape de Verd Islands

After having been twice driven back by heavy south-western gales, Her Majesty's ship "Beagle," a ten-gun brig, under the command of Captain Fitz Roy*, R.N., sailed from Devonport on the 27th of December, 1831. The object of the expedition was to complete the survey of Patagonia* and Tierra del Fuego*, commenced under Captain King* in 1826 to 1830– to survey the shores of Chile*, Peru*, and of some islands in the Pacific*– and to carry a chain of chronometrical measurements round the World. On the 6th of January we reached Teneriffe, but were prevented landing, by fears of our bringing the cholera: the next morning we saw the sun rise behind the rugged outline of the Grand Canary Island, and suddenly illumine the Peak of Teneriffe, whilst the lower parts were veiled in fleecy clouds.

The neighbourhood of Porto Praya, viewed from the sea, wears a desolate aspect. The volcanic fires of a past age, and the scorching heat of a tropical sun, have in most places rendered the soil unfit for vegetation. The scene, as beheld through the hazy atmosphere of this climate, is one of great interest; if, indeed, a person, fresh from sea, and who has just walked, for the first time, in a grove of cocoa-nut trees, can be a judge of anything but his own happiness. The island would generally be considered as very uninteresting, but to any one accustomed only to an English landscape, the novel aspect of an utterly sterile land possesses a grandeur which more vegetation might spoil. I state this on the authority of Dr. E. Dieffenbach, in his German translation of the first edition of this Journal), the reckless destruction of which has caused here, as at St. Helena*, and at some of the Canary islands, almost entire sterility. It is brightly coloured, but not so beautiful as the European species: in its flight, manners, and place of habitation, which is generally in the driest valley, there is also a wide difference.

One day, two of the officers and myself rode to Ribeira Grande, a village a few miles eastward of Porto Praya. Until we reached the valley of St. Martin, the country presented its usual dull brown appearance; but here, a very small rill of water produces a most refreshing margin of luxuriant vegetation. Having procured a black Padre for a guide, and a Spaniard who had served in the Peninsular war as an interpreter, we visited a collection of buildings, of which an ancient church formed the principal part.

We returned to the Vnda to eat our dinners. We presented the black priest with a few shillings, and the Spaniard, patting him on the head, said, with much candour, he thought his colour made no great difference.

Another day we rode to the village of St. Domingo, situated near the centre of the island. On a small plain which we crossed, a few stunted acacias were growing; their tops had been bent by the steady trade-wind*, in a singular manner– some of them even at right angles to their trunks. The direction of the branches was exactly north-east by north, and south-west by south, and these natural vanes must indicate the prevailing direction of the force of the trade-wind*.

Near Fuentes we saw a large flock of guinea-fowl– probably fifty or sixty in number.

The scenery of St. Domingo possesses a beauty totally unexpected, from the prevalent gloomy character of the rest of the island.

One morning the view was singularly clear; the distant mountains being projected with the sharpest outline, on a heavy bank of dark blue clouds.

Figure C.6: First section in the first chapter of *The Voyages of the Beagle*, summarised at 30% of its original size (continued in next figure).

Generally the atmosphere is hazy; and this is caused by the falling of impalpably fine dust, which was found to have slightly injured the astronomical instruments. From the direction of the wind whenever it has fallen, and from its having always fallen during those months when the harmattan is known to raise clouds of dust high into the atmosphere, we may feel sure that it all comes from Africa*. It is, however, a very singular fact, that, although Professor Ehrenberg* knows many species of infusoria peculiar to Africa*, he finds none of these in the dust which I sent him.

The geology of this island is the most interesting part of its natural history. It rests on ancient volcanic rocks, and has been covered by a stream of basalt, which must have entered the sea when the white shelly bed was lying at the bottom.

During our stay, I observed the habits of some marine animals. On each side of the lower surface, or foot, there is a broad membrane, which appears sometimes to act as a ventilator, in causing a current of water to flow over the dorsal branchiae or lungs. Besides this means of defence, an acrid secretion, which is spread over its body, causes a sharp, stinging sensation, similar to that produced by the Physalia, or Portuguese man-of-war.

I was much interested, on several occasions, by watching the habits of an Octopus, or cuttle-fish. By means of their long arms and suckers, they could drag their bodies into very narrow crevices; and when thus fixed, it required great force to remove them. They appear to vary their tints according to the nature of the ground over which they pass: when in deep water, their general shade was brownish purple, but when placed on the land, or in shallow water, this dark tint changed into one of a yellowish green.

This cuttle-fish displayed its chameleon-like power both during the act of swimming and whilst remaining stationary at the bottom. Remaining for a time motionless, it would then stealthily advance an inch or two, like a cat after a mouse; sometimes changing its colour: it thus proceeded, till having gained a deeper part, it darted away, leaving a dusky train of ink to hide the hole into which it had crawled.

While looking for marine animals, with my head about two feet above the rocky shore, I was more than once saluted by a jet of water, accompanied by a slight grating noise. At first I could not think what it was, but afterwards I found out that it was this cuttle-fish, which, though concealed in a hole, thus often led me to its discovery.

Figure C.7: First section in the first chapter of *The Voyages of the Beagle*, summarised at 30% of its original size (continued from the previous figure).

Rio

Rio, a Brazilian city in the Southeast coast of the country, has been known as one of the most beautiful cities on Earth. Meanwhile, one should note that the concept of a city is a bit bigger in Rio than in other nice small cities, like Rome or Paris. Rio has 5.6 million inhabitants in the city and around 10 million inhabitants in the wide region around it. Many people from Sao Paulo, Mexico City, and Tokyo used to say Rio is a very nice little town to expend the weekend.

Rio is a very large city, with some prejudice against people wearing shorts with flowers, t-shirts with flowers, gold jewelry and a using camcorders. These people, which usually call themselves tourists, have the local denomination of "fool," "idiot," or "duck". Just ignore it.

Rio is a place to find people. "You do not bring a sandwich to a banquet" is a common saying there.

Rio is the Cídade Maravilhosa (Marvellous City). Jammed into the world's most beautiful setting - between ocean and escarpment - are seven million Cariocas, as Rio's inhabitants are called. The Cariocas pursue pleasure like no other people: beaches and the body beautiful; samba and beer; football and the local firewater, cachaa rum.

Rio has its share of problems: a third of the people live in the favelas shanty towns that blanket many of the hillsides; the poor have no schools, no doctors and no jobs; drug abuse and violence are endemic; and police corruption and brutality are commonplace... Rio 's reputation as a violent city caused a sharp reduction in tourism in the 1990s, but travelers will find themselves no more at risk than in most large cities in the world.

Rio is divided into a zona norte (northern zone) and a zona sul (southern zone) by the Serra da Carioca, steep mountains that are part of the Parque Nacional da Tijuca. The view from the top of Corcovado, the 750m 2460ft mountain peak with the statue of Christ the Redeemer at its summit, offers the best way to become geographically familiar with the city. Favelas crowd against the hillsides on both sides of town.

Rio's famous glitzy Carnaval is a fantastic spectacle, but there are more authentic celebrations held elsewhere in Brazil. In many ways, Carnaval can be the worst time to be in Rio. Everyone gets a bit unglued at this time of year: taxi fares quadruple, accommodation triples and masses of visitors descend on the city to get drunk, get high and exchange exotic diseases.

Rio's exceptional geographical location and its proximity to the country's greatest industrial centers made it an attraction point for large migratory flows of the poor population. In 1763, Rio de Janeiro displaced Salvador da Bahia as the colonial capital of Brazil.

In 1763, Rio displaced Salvador de Bahia as the colonial capital of Brazil, and for the next two centuries the city went unchallenged as the urban center of Brazil.

Even though the growth of competing metropolitan regions has eroded its leading edge, Rio continues to be an important urban centre. Industry has continued to grow. The clothing industry and pharmaceutical as well as the medical and food industries are highly important.

Rio's carnival is the most important one, known all over the world, and it attracts a huge amount of tourists from several countries.

Rio is eleven miles long –and in most of the best neighborhoods not much more than a half mile to a mile wide. To explain this odd anomaly, consider that Rio is a city pressed between a steep mountain and the ocean. The city is long and narrow. The best neighborhoods for anyone wanting go buy real estate are Flamego, Jardim Botanico, Ipanema, Leblon, Copacabana, Lagoa and Santa Teresa. My favorites have always been Flamego and Santa Teresa. Flamengo would appeal to most readers, but Santa Teresa will only appeal to the more adventuresome.

Figure C.8: Page about the concept *Río*, referring to Rio de Janeiro, in Brazil. Note that *Río* is the Spanish and Portuguese for *river*, and even so all the information collected referred to the right meaning of *Río* (16 more paragraphs remaining).

Cornwall

England's Cornwall Walk Cornwall and thew Lizard Peninsula are considered England's finest coastal trails. If you are a lover of nature, the sea, history, photography or romance, Cornwall is your walk. This area is reminiscent of the Mediterranean, but the steep-sided harbors, fishing villages, flocks of sea birds, purple heather and yellow gorse make it more interesting and dramatic.

Cornwall's ancient kings were powerful, accruing wealth through trading minerals and escaping involvement in most of the wars and invasions that troubled the rest of Britain. Legendary names like King Mark and King Tristian of 'Tristian and Iseult' fame did actually reign from Cornwall and there is evidence of the presence of a King Arthur who lived over 1000 years ago. The myth of King Arthur and the Knights of the Round Table and the fabled kingdom of Camelot was written so many centuries later that it is impossible to rely on the story, but there was a King Arthur who caused rather a headache for the Saxons around AD500 beating them in 12 battles. However, there is nothing to disprove the notion that Tintagel was the site of his castle and the ruins to be found there on a breathtaking fortress of rock have an eerie credibility to them...

Cornwall still kept its character by not adopting English place names with -ton, etc, and retained its own language and customs. At the turn of the century, the Cornish were making stone crosses, not in total respect of Christianity, but as a symbols of Celtic nationalism. Whoever took control of England, the Cornish would fight. After the Saxons came the Normans in 1066 and again the Cornish fought them off. By now, the Cornish were feared by all and pretty much considered a lawless and barbarous bunch. This reputation persisted until the end of the last century -when Cornwall was occasionally referred to in newspapers as West Barbary.

Cornwall sided with the Royalists during the Civil War and were one of the last places in the British Isles to be run over by Cromwell's New Model Arrny.

Cornwall was becoming quite prosperous at this time -mining, agriculture and fishing all created huge wealth for the gentry and represented big bucks for the crown. Pilchards were big business and the ports of Newlyn and Charlestown, in particular, would have hundreds of fishing boats in their fleets ; entire communities depended on the fortunes of the fishermen. Many fishing ports had huts overlooking the bay for shoals of pilchards as they came in. A 'huer' would man the hut and shout upon sight of a shoal after which there would be an almighty scramble to get to the water to bring in the booty.

From 1700 onwards Cornwall was a main player in the Industrial Revolution. 90 of the world's tin came from Cornish mines and the county also produced copper, arsenic, lead and silver. The need to mine deeper, further and faster prompted several significant developments in the new field of engineering. The first powered vehicle made its debut in Camborne, a locomotive designed by Richard Trevithick. Forget Stephenson and his Rocket, he merely picked up where Trevithick left off and claimed all the credit when our man moved onto developing steam power on ships after a couple of mishaps with prototype engines. Gas lighting and the miners safety lamp were also invented in Camborne.

(thirteen more paragraphs remaining)

Figure C.9: Page about the concept *Cornwall.*

Appendix D

Introducción

La World Wide Web ha popularizado el uso de hipermedia para transmitir y compartir información. Sin embargo, la gran cantidad de datos disponible en Internet crece continuamente, lo que puede provocar problemas para los usuarios [Wu and de Bra, 2002]: en primer lugar, los sitios web estáticos ofrecen la misma información a todos los usuarios independientemente de sus intereses (el así llamado paradigma de *"tamaño único"*, *"one size fits all"*). En esta situación, a los usuarios les puede resultar muy difícil encontrar la información relevante, o han de dedicar mucho tiempo a revisar datos sin interés hasta que la encuentran. En segundo lugar, las páginas web también pueden provocar problemas de comprensión, dado que el autor de las páginas ha hecho presuposiciones implícitas sobre los conocimientos previos de los usuarios. De este modo, un usuario puede encontrarse con una página que contiene conocimientos demasiado básicos o, por el contrario, con una página que no es capaz de comprender porque le falta la formación necesaria.

Esto ha abierto líneas en varias áreas diferentes. Por un lado tenemos las aplicaciones de *Hipermedia Adaptativa*, que intentan proporcionar a los usuarios la información que necesitan y ayudarles a navegar entre los documentos. Son ejemplos las aplicaciones de Recuperación de Información [Baeza-Yates and Ribeiro-Neto, 1999], que buscan información en enormes almacenes de documentos o en Internet según el perfil o la consulta de un usuario; los sistemas adaptativos de educación basados en hipermedia, que ayudan a los estudiantes a navegar por un curso en función de los conocimientos que vayan aprendiendo; o los sistemas de información en línea, que intentan proporcionar a los usuarios la información que necesitan sobre un tema en particular, de acuerdo con sus intereses y el contexto en que se encuentran (como, por ejemplo, los sistemas de información en los museos).

Por otra parte, ha surgido otro grupo de aplicaciones que puede mejorar el problema de la sobrecarga de información, a partir de la investigación en el Procesamiento de Lenguaje Natural. Aquí podemos citar las aplicaciones de Extracción de Información, que obtienen información estructurada a partir de textos; los sistemas automáticos de Respuesta, que buscan la respuesta a la pregunta formulada por un usuario en una colección de documentos; o las aplicaciones de Resúmenes Automáticos, que condensan la información relevante encontrada en una o varias fuentes de texto. En la mayoría de los casos, los sistemas de Recuperación de Información utilizan herramientas del campo de la lingüística computacional, tales como los analizadores morfológicos, que obtienen el lexema de las palabras. Por tanto, podemos también incluir estos sistemas en este grupo de aplicaciones.

La iniciativa denominada *web semántica* [Berners-Lee et al., 2001] procura definir lenguajes estándares de representación de conocimiento, tales como RDF, DAML+OIL o OWL, para permitir que las páginas de

hipertexto incluyan descripciones de su contenido. De esta manera, las páginas web incluirían información semántica que indique a qué conceptos se refieren, como personas, libros, películas, etc. Hasta que se definan estos lenguajes y servicios, es posible utilizar técnicas automáticas para discriminar entre las páginas web. Por ejemplo, la desambiguación del sentido de las palabras consiste en averiguar con qué significado se está utilizando una palabra en un contexto dado [Ide and Véronis, 1998]. Utilizando procedimientos automáticos, quizá se podría comprobar si las palabras de la consulta del usuario se refieren al mismo concepto que las palabras de un documento web, para localizar los que son más relevantes para el usuario.

El exceso de información es un grave problema y se están explorando muchas soluciones posibles para manejarlo. No sólo hay cantidades enormes de información en Internet; el problema también aparece, en mayor o menor grado, en los ordenadores personales. La capacidad actual de los sistemas de almacenamiento permite almacenar enormes volúmenes de información, y una búsqueda para encontrar datos puede requerir varios minutos. Se pueden realizar muchas aplicaciones para ayudar al usuario a encontrar información relevante. Los siguientes son algunos ejemplos de posibles aplicaciones, algunos de los cuales ya existen como prototipos o aplicaciones comerciales, mientras que otros podrían aparecer en el futuro:

- Extraer y estructurar información relevante de documentos que tratan sobre un tema específico, o tienen una estructura similar, como buscar información en artículos de periódico o comparar currículos con anuncios de trabajo.

- Filtrar y resaltar información relevante, por ejemplo, eliminando mensajes basura y anuncios del buzón de correo electrónico, o seleccionando documentos que parecen particularmente importantes para el receptor de una lista de distribución.

- Extraer y estructurar información relevante (tal como números de teléfono, direcciones o informes) a partir de datos heterogéneos, como mensajes de correo electrónico, para que sea más fácil encontrarlos.

- Sistemas de planificación que toman decisiones y las llevan a cabo para satisfacer una necesidad del usuario. Por ejemplo, si un usuario quiere asistir a una conferencia, un sistema de planificación podría obtener de Internet información acerca de la fecha de la conferencia, vuelos y hoteles disponibles, y generar un plan de viaje; al mismo tiempo, revisaría la agenda personal del usuario para aplazar las tareas que hubiese previstas en esas fechas.

- Sistemas que responden a las preguntas del usuario, aunque para ello sea necesario razonar con la información de una base de conocimiento, como hacer deducciones de sentido común o demostrar un teorema.

Este trabajo se centra en la construcción automática de sitios hipermedia adaptados a las necesidades de los usuarios particulares. Las siguientes secciones describen en detalle la tarea que se aborda: para comenzar, las motivaciones para el trabajo se describen en la sección D.1. A continuación, la sección D.2 describe los requisitos del resultado esperado. Las secciones D.3 y D.4 describen el diseño del entorno teórico y los diferentes módulos de la aplicación práctica que se ha construido de acuerdo con la arquitectura. La sección D.5 describe los recursos de texto que se han usado como datos de entrenamiento o de prueba para los distintos módulos, y los textos a partir de los cuales se han construido automáticamente sitios web adaptativos. Finalmente, las secciones D.6 y D.7 contienen, respectivamente, un resumen de las contribuciones de este trabajo y la visión general de la tesis.

D.1 Motivación

Como se dijo anteriormente, la hipermedia adaptativa apareció para evitar que usuarios con diferentes características hayan de recibir la misma información, independientemente de sus características. La adaptación suele utilizar un *modelo de usuario* que contiene algunas de sus características, tales como sus preferencias o conocimiento previo, y un *modelo de dispositivo*, que contiene las características del dispositivo usado, tales como el tamaño de la pantalla o la velocidad de la conexión a la red. Estos modelos se usan para proporcionar información personalizada. Algunas de las técnicas más usadas incluyen el *soporte para la navegación adaptativa*, que consiste en adaptar la estructura del hiperespacio (los enlaces) para guiar al usuario hacia la información más interesante, y la *presentación adaptativa*, que consiste en adaptar los contenidos de las páginas web a las necesidades del usuario, por ejemplo, ocultando los párrafos irrelevantes o resaltando aquello que el sistema cree que será más relevante [de Bra et al., 1999a].

Estas técnicas, al tiempo que simplifican la labor del visitante de la web, aumentan grandemente el trabajo del autor de la misma. A la hora de construir una web adaptativa, no basta con escribir los contenidos y conectar las páginas entre sí, sino que además es necesario definir las características de los usuarios que han de modelizarse y las reglas que determinarán cómo esas características cambiarán los contenidos y guiarán a los usuarios. Una posibilidad consiste en crear herramientas de autor para hipermedia adaptativa [Brusilovsly et al., 1998, Murray et al., 2000, Sanrach and Grandbastien, 2000]. Éstas facilitan la labor proporcionando un entorno en el que varias características están más o menos fijadas de antemano, tales como los perfiles de usuario o los mecanismos de adaptación, pero el diseñador aún ha de dedicar mucho tiempo para explorar y aprovechar todas las posibilidades de adaptación de los sistemas.

D.1.1 Problema

Uno de los problemas del exceso de información disponible es la falta de tiempo para consultarla toda. Los documentos suelen ser grandes y multidisciplinares, e incluyen párrafos y secciones sobre diferentes temas. Un usuario que no disponga de mucho tiempo y necesite unos conocimientos sobre un tema en particular puede, posiblemente, descubrir que dichos conocimientos están dispersos en diversas secciones de varios libros. En esta situación particular, sería útil disponer de un procedimiento automático para seleccionar información de fuentes dispares, juntarla toda, de acuerdo con el perfil de algún usuario, y proporcionar una estructura interna con secciones separadas e hiperenlaces entre las secciones.

Por ejemplo, imaginemos que alguien necesita aprender sobre la interfaz entre dos programas diferentes, tales como el lenguaje Java y el gestor de base de datos relacional `mysql`. Si todo el trabajo se realizase manualmente, el procedimiento sería el siguiente: primero, identificaría algunos documentos sobre el tema, como un manual de programación para Java, la guía de usuario de `mysql`, y el manual de uso del driver JDBC para `mysql`. A continuación, estudiando los índices de los distintos documentos, decidiría cuáles son las secciones más relevantes para sus propósitos, como las porciones del manual de Java que se refieren a la utilización de bases de datos (tales como la que describe el paquete `java.sql`); las porciones del manual de `mysql` que explican las interfaces con los lenguajes de programación; y el manual del driver de JDBC. Tras esta selección, procedería a leer las secciones relevantes. Es posible que completara la información con datos adicionales, posiblemente buscados en Internet, con documentos de preguntas frecuentes, y con listas de discusión de usuarios que tuvieron problemas usando esas herramientas.

El trabajo propuesto en esta tesis tiene como finalidad reproducir las acciones que realizaría un ser humano en una situación similar, cuando buscase información para satisfacer una necesidad particular:

1. La información puede proceder de fuentes sobre algún campo de aplicación específico, como libros y artículos sobre el tema en cuestión, que son suministrados por el usuario. Es de esperar que esos documentos contengan alguna información muy relevante, dado que han sido seleccionados cuidadosamente por el usuario.

2. Los datos seleccionados también se pueden extender con información recogida automáticamente de documentos de propósito general, como los existentes en Internet.

3. Esta información puede ser mostrada al usuario de manera estructurada, con herramientas que ayuden a reducir la información manteniendo tan sólo los fragmentos más relevantes, y que ayuden a navegar a través de esa información.

Para construir esta herramienta, ha sido necesario investigar lo siguiente:

- Cómo representar y adquirir los intereses del usuario.
- Cómo seleccionar la información más relevante.
- Cómo estructurar y organizar la información en un sitio web adaptativo.

D.1.2 Motivación lingüística

Para crear el sitio web adaptativo a partir de un conjunto de documentos es necesario analizar, hasta cierto punto, la información proporcionada en esos textos. Aunque algunas tareas, tales como los cálculos matemáticos, tradicionalmente han sido relativamente más fáciles de automatizar con un ordenador, otros, especialmente los que requieren la codificación y aplicación de grandes cantidades de conocimiento, aún han de ser realizados por seres humanos. Dado lo costoso que resulta codificar esta información, se dice que tienen el problema de la adquisición del conocimiento. Los problemas que conlleva el procesamiento de textos sin restricciones son un ejemplo típico de estas tareas difíciles, puesto que generalmente es necesario codificar grandes cantidades de conocimiento lingüístico o de sentido común y, en general, sólo funcionan en campos de aplicación restringidos.

Por otra parte, en los últimos años han visto la luz varias aplicaciones que realizan análisis de textos, tales como aplicaciones de recuperación de información para motores de búsqueda, o sistemas de resúmenes automáticos para procesadores de textos. Aunque estos aún tienen cierto número de errores, es de esperar que su exactitud mejore conforme progresa la investigación.

Con respecto al procesamiento de lenguaje natural, algunos problemas han sido o son especialmente difíciles. Uno de ellos es el del análisis sintáctico, que en los últimos años ha recibido mucha atención. Actualmente hay aplicaciones que obtienen resultados razonablemente buenos, al menos para el inglés y otros idiomas bien estudiados. La disponibilidad de textos anotados con información sintáctica resulta especialmente útil, dado que ha sido posible construir analizadores que utilizan técnicas de aprendizaje automático [Hockenmaier and Steedman, 2002, Xia, 1999]. Sin embargo, aún queda trabajo por hacer, por ejemplo, aumentar la robustez del análisis de frases mal construidas, o construir analizadores para lenguajes minoritarios y sin recursos.

En lo que respecta al análisis semántico, la disponibilidad de analizadores sintácticos ha permitido estudiar con más detalle técnicas para descubrir y representar el significado de las oraciones. Las redes semánticas conceptuales, como WordNet, también han resultado especialmente útiles [Miller, 1995]. Actualmente se

está estudiando también, entre otras cosas, la construcción de interfaces de diálogo, y de herramientas que realicen análisis pragmático de los textos, que necesitan de un buen analizador semántico para tener buenos resultados.

Muchas de las aplicaciones del procesamiento lingüístico se pueden usar para hipermedia adaptativa. Estas incluyen las interfaces de diálogo, de respuesta a preguntas (sistemas que buscan las respuestas a una pregunta de un usuario en una colección de documentos), de resúmenes automáticos, de generación de lenguaje natural, y muchas otras. Esta es la razón por la cual la investigación en lingüística computacional es también una motivación importante para este trabajo.

D.1.3 Resultados esperados

El exceso de datos en Internet fuerza a los usuarios a pasar mucho tiempo buscando información. Varios estudios concluyen que los empleados de las empresas dedican gran parte de su tiempo de trabajo a realizar búsquedas en Internet. Por ejemplo, McKinley [1997] señala que posiblemente el 20% del trabajo administrativo se dedica a recoger y almacenar documentos importantes; y, de acuerdo con una encuesta hecha pública por Mediapps el 20 de Octubre de 2000, los empleados dedican miles de horas a buscar información relevante en Internet, lo que les cuesta a las compañías decenas de miles de libras esterlinas al año[1]. Considerando este hecho, un sistema adaptativo que ayude a un usuario a buscar información acerca de un tema de interés, ya sea en una pequeña colección de documentos o en Internet, podría ser muy útil, dado que reduce el tiempo necesario para encontrar la información, y el coste para las empresas es menor.

En la mayor parte de los casos, se puede dividir en dos pasos el diseño de sitios hipermedia adaptativos. En primer lugar hay un paso fuera de línea, durante el cual se recoge y se estructura la información relevante. Puede ser necesario, en este paso, crear una base de conocimiento acerca de los conceptos descritos en los textos, de manera que pueda usarse para saber en todo momento cuáles de estos conceptos han sido ya visitados por el usuario y cuáles no. Este paso incluye las siguientes tareas:

- *La identificación de los temas relevantes y las secciones del sitio hipermedia.* Si el autor ya es un experto, este paso no ha de ser especialmente trabajoso; en caso contrario, podría ser necesaria una colaboración con expertos para recoger la información. Esta tarea es común al diseño de hipermedia no adaptativa.

- *Escribir los contenidos de las secciones.* El autor ha de escribir unidades textuales que describan los diversos temas. En contraste con las web estáticas tradicionales, puede ser necesario escribir varias versiones de los mismos textos, por ejemplo, en diferentes idiomas, con diferentes longitudes, o para lectores con niveles culturales diferentes.

- *Generar la base de conocimiento*, en el caso de que sea necesaria para la aplicación, indicando qué conceptos aparecen en cada porción de texto y las relaciones entre los textos o entre las páginas. Por ejemplo, un texto podría contener la descripción de algo más específico que otro texto; en este caso, podría ser necesario añadir la restricción de que el usuario ha de leer el segundo texto antes de poder leer el primero.

A continuación, hay un paso en línea que se ejecuta cuando los usuarios acceden al sitio adaptativo para buscar información. Durante este paso, es necesario realizar los siguientes pasos:

[1] http://web01.mediapps.com/web/uk.nsf/$$pagesweb/CompanyPressReleases2

- *Adaptar los contenidos* al perfil del usuario o a su entorno. Esto puede incluir mostrar fragmentos de texto diferentes (por ejemplo, en distintos idiomas), cambiar el formato (por ejemplo, sustituir textos por imágenes o vídeo que proporcionan la misma información), y otras muchas técnicas.

- *Adaptar la estructura del sitio hipermedia adaptativo* para guiar al usuario en su búsqueda de información. Esto incluye resaltar u ocultar enlaces si se consideran especialmente interesantes o irrelevantes; anotar los enlaces para que el usuario sepa a dónde conducen, crear enlaces nuevos sobre la marcha; etc.

> El objetivo último de esta tesis es la automatización completa de todos estos pasos, para que el trabajo del diseñador quede reducido a una supervisión del sistema. Usando un conjunto de textos que contienen los contenidos del futuro sitio hipermedia, el sistema ha de identificar las secciones relevantes, seleccionar el texto que aparecerá en cada una de ellas, estructurar las secciones en un hiper-grafo, y proporcionar los mecanismos de adaptación para los distintos usuarios.

El trabajo realizado obtiene dos resultados. En primer lugar, la descripción de una arquitectura que combina ideas de diversas áreas para llevar a cabo el objetivo de automatizar la creación de sitios web adaptativos. En segundo lugar, la implementación de la arquitectura en un sistema llamado WELKIN. En el futuro, usaremos el nombre WELKIN para referirnos a la implementación particular del entorno. El Apéndice C muestra algunas páginas de ejemplo generadas automáticamente para distintos usuarios. Las secciones siguientes describen estas tareas con más detalle, junto con las hipótesis iniciales.

D.2 Objetivos generales de la investigación

Esta sección describe los requisitos que se definieron para este trabajo. Dado el estado actual de la tecnología, si la arquitectura fuese demasiado general, podría resultar imposible construirla, de modo que es necesario restringir las características y el alcance de dicha arquitectura.

Requisitos operativos

El objetivo general es la identificación y la presentación de información relevante para el usuario, dada la sobrecarga de información existente. Los datos relevantes deberían estar estructurados como un sitio hipermedia.

El material inicial consta de los siguientes elementos:

1. Uno o varios textos, escogidos por los usuarios, acerca un tema que es relevante para ellos. Por ejemplo, un usuario interesado en zoología y en los orígenes de la teoría de la evolución de Darwin podría seleccionar los libros de Darwin como datos de partida.
2. Una conexión a Internet, para recoger información adicional.
3. Una descripción de los intereses del usuario.

El resultado es un sitio hipermedia adaptativo completo, construido a partir de la información encontrada en los documentos originales y en Internet. El sitio debería mostrar a cada usuario solamente la información particular que se considere relevante, de acuerdo con la representación interna de los intereses del usuario,

estructurada internamente con secciones e hiperenlaces entre ellos. La estructura del hiperespacio debería ayudar a los usuarios a encontrar los datos útiles.

Los requisitos operativos que debería satisfacer este sitio web generado son los siguientes:

- Proporcionar una manera sencilla para que los usuarios describan sus intereses.
- Proporcionar información relevante.
- Esta información debería estar estructurada como un sitio web completo, con enlaces apropiados para navegar por la información.
- También debería proporcionar mecanismos para adaptar la información a distintos usuarios, o a un mismo usuario con diferentes requisitos, tales como un mayor o menor nivel de compresión de la información.

Objetivos de la adaptación Las fuentes de conocimiento multidisciplinar contienen información relevante para distintas áreas de conocimiento. A menudo, un usuario ha de utilizar un texto multidisciplinar del que sólo está interesado en una pequeña parte. La adaptación de contenidos será una característica muy importante de la arquitectura. La información original será anotada con la información necesaria para que pueda ser seleccionada fácilmente para presentarla a los usuarios interesados en ella.

Los propósitos de la adaptación se pueden resumir en los siguientes puntos:

- Debería ayudar a cada usuario a separar la información relevante de la irrelevante, creando un modelo de usuario que refleje sus intereses, y resaltar las porciones del texto más relevantes.
- Debería adaptar la presentación del texto a las necesidades del usuario, adaptando la cantidad de información presentada al tiempo que se tenga disponible.

Modelos de usuario La arquitectura tiene que codificar información acerca de los intereses y objetivos del perfil del usuario. En algunos campos, como la Recuperación de Información o las respuestas automáticas, lo único que conoce la aplicación acerca del usuario es una consulta. Cuando el usuario escribe una segunda consulta, la primera se olvida, de manera que el perfil del usuario cambia completamente cada vez que utiliza el sistema. Por otra parte, los sistemas de información en línea generalmente almacenan características más estáticas del usuario, como sus conocimientos, sus intereses generales o su idioma preferido.

En este caso particular, el usuario quiere encontrar cantidades relativamente grandes (lo bastante para crear un sitio web completo) dentro de un depósito de información aún mayor. El resultado del sistema será un sitio web que requiere cierto tiempo para leerlo. Por tanto, el modelo de usuario ha de almacenar características más estables de los usuarios, tales como sus intereses.

Por otro lado, el sitio web generado ha de satisfacer alguna necesidad temporal del usuario, y es probable que cuando éste haya leído el sitio web entero el perfil del usuario ya no sea necesario. En otras palabras, las características del usuario, tales como sus conocimientos previos, las páginas que ya ha visitado, o los resultados de ejercicios para comprobar si realmente ha asimilado la información, no son de interés y no es necesario modelarlos. Una vez que se ha generado el sitio de información en línea, los usuarios son libres para navegar por él, y no se tendrá control sobre si realmente aprenden toda la información disponible, como la hay en los sistemas educativos.

Técnicas de adquisición del modelo de usuario La función principal del modelo de usuario es almacenar su interés sobre los diversos temas que puedan aparecer en los documentos. Hay varias maneras de

adquirir estos datos:

- Con un conjunto de temas predefinidos (llamados estereotipos), tales como las principales divisiones de las ciencias y las artes. En este caso, un usuario podría escoger uno o varios de los intereses predefinidos. Por ejemplo, cuando se dé de alta en el sistema, un usuario podría indicar que está interesado en historia o en informática.

- Pidiendo al usuario que proporcione un conjunto de documentos o párrafos que considera relevantes. De este modo, el usuario podría tener desde el comienzo un perfil personalizado.

En cualquier caso, es deseable que el usuario o el sistema puedan modificar dinámicamente el perfil de intereses, dependiendo de las acciones realizadas, las páginas visitadas y las cosas por las que se ha mostrado interés.

Modelos de adaptación

Muchas aplicaciones de hipermedia adaptativa incluyen tres sub-modelos [Wu and de Bra, 2002]: el *modelo de dominio*, que contiene información sobre los contenidos del sitio (tales como fragmentos de textos o imágenes) y su estructura (las relaciones entre las secciones); el *modelo de usuario* contiene información acerca del usuario (esto incluye el dispositivo que esté usando); y el *modelo de adaptación* contiene las reglas que deciden cómo mostrar los contenidos, y qué contenidos mostrar a cada usuario.

A este respecto, el principal propósito de la arquitectura propuesta es la automatización del modelo de dominio: los contenidos del sitio hipermedia y las reglas que definen qué contenidos presentar y en qué orden. Esto tiene implicaciones sobre el *modelo de usuario* —porque los contenidos dependen de los intereses y los objetivos— y sobre el *modelo de adaptación*, porque el diseño del sitio depende, hasta cierto punto, de las técnicas de adaptación utilizadas.

Áreas de aplicación

La generación de cualquier clase de textos sobre cualquier tema no es el objetivo de este trabajo. Diferentes áreas de conocimiento usan terminologías diferentes; los documentos se escriben con estructuras diferentes; pueden incluir tablas, figuras, ejemplos, código o pseudocódigo de programas, diagramas y muchos otros tipos de información, cada uno de los cuales habría de ser procesado con herramientas diferentes.

Por otro lado, es posible definir una arquitectura genérica. En este entorno, diferentes módulos podrían *enchufarse* para que cada uno de ellos realice un trabajo muy específico, y su resultado podría ser utilizado por el generador de hipermedia para decidir si cada porción de información es o no es útil para algún usuario. Por ejemplo, la terminología relevante en informática consistiría en palabras técnicas; mientras en biología incluiría nombres comunes de animales y plantas, nombres científicos, nombres de compuestos orgánicos, e incluso lugares y fechas; en metafísica, tendrían un puesto relevante los nombres que se refieren a abstracciones. Por tanto, los módulos que reconocen términos desconocidos deberían ser sustituibles, para el caso de que el sistema se utilice con diferentes textos. Las siguientes ideas describen una solución posible:

- La arquitectura ha de ser modular y los diferentes módulos han de ser intercambiables.

- Ha de proporcionar un mecanismo para que los módulos se comuniquen información entre sí dentro de la arquitectura global, por ejemplo, añadiendo anotaciones en los documentos.

- Ha de haber una manera de indicar si algún módulo necesita las anotaciones producidas por otro módulo. Por ejemplo, un módulo que identifica términos relevantes podría necesitar que otro módulo haya identificado previamente los sintagmas nominales en los textos.
- Para distintos tipos de textos, debería ser posible definir módulos alternativos que realicen la misma función, pero especializados en campos de aplicación diferentes, como reconocer términos de interés en textos sobre biología o informática, o procesar textos en lenguajes diferentes.

Requisitos de la interfaz

El resultado es una colección de secciones con la estructura de un sitio hipermedia. Considerando que el entorno hipermedia más popular hoy día es la World Wide Web, este resultado debería generarse en HTML, que puede visualizarse desde cualquier navegador web estándar, desde cualquier ordenador conectado a Internet. Como en cualquier otra aplicación hipermedia, la información estará estructurada en forma de hiperdocumentos y enlaces entre ellos. Además, el sistema usará algunos de los métodos y técnicas usuales del campo de la hipermedia adaptativa, tales como los siguientes:

- **Contenidos adaptativos**, mediante la adición o eliminación de fragmentos de texto; reduciendo o expandiendo la información en función de los deseos del usuario; y utilizando técnicas de Procesamiento de Lenguaje Natural para producir resúmenes de los textos.
- **Soporte a la navegación adaptativa**, mediante *adaptación de enlaces* (creando enlaces nuevos de acuerdo con el perfil del usuario); y mediante *la anotación de enlaces* con colores, de acuerdo con el tipo de información al que conducen.

D.3 Arquitectura: descripción teórica

Esta sección describe una manera en la que se pueden satisfacer los requisitos descritos en la sección anterior mediante una arquitectura. El procesamiento se ha dividido en dos partes, tal como se indicó en la sección D.1. La primera, realizada fuera de línea, recoge todo el material necesario para saber cuáles son los textos que son importantes para cada uno de los posibles intereses del usuario. El segundo paso, en línea, incluye todas las acciones que deben llevarse a cabo cuando un usuario accede al sistema buscando información.

D.3.1 Identificación de contenidos (fuera de línea)

El primer paso consiste en decidir cuál es la información que contendrá el sitio web. Dado un tema, suele ser relativamente sencillo encontrar textos, a menudo lineales, que contengan datos sobre ese tema. *La primera hipótesis* que se adoptará es que hay textos disponibles acerca del tema del futuro sitio web. A continuación, habrá que decidir cómo estructurar la información de esas fuentes como un sitio hipermedia.

En segundo lugar, para todos los temas de interés suele haber una terminología específica que no se usa en el lenguaje común (y que, por tanto, no aparece en los diccionarios de propósito general). En un texto sobre historia, por ejemplo, esta terminología incluye nombres de personajes, lugares, partidos políticos y tendencias, países, etc.; en un texto sobre matemáticas, incluye los nombres de teorías, teoremas, entidades algebraicas, etc. *La segunda hipótesis* adoptada en este trabajo es que se puede construir un sitio web completo a partir de la información de los textos lineales estructurándola del siguiente modo:

- Con páginas de hipertexto para cada una de las secciones relevantes que aparezcan en los textos originales.
- Con páginas de hipertexto que describan la terminología específica y los términos de propósito general que se usen con mucha frecuencia.
- Páginas de índice que incluyan los términos identificados, las secciones, etc.

Por ejemplo, si existe una memoria de investigación de un departamento universitario, estructurada de acuerdo con ciertas normas internas, los siguientes hiperdocumentos se podrían generar automáticamente a partir de ella:

- Páginas web que resuman cada una de las secciones del informe: la introducción, los distintos tipos de investigación realizados, la financiación, las conclusiones, etc.
- Páginas diferentes para cada uno de los términos específicos, tales como los nombres de los temas de estudio, los nombres de los investigadores, los lugares donde se han realizado estancias de investigación, etc.
- Páginas de índice de todo lo anterior: listas de personal, de áreas de investigación, etc.

Por supuesto, esta hipótesis restringe hasta cierto punto la información que puede generadarse automáticamente. Por ejemplo, no permite generar páginas como mapas del sitio web, o las que describan más de un término. Sin embargo, se supondrá que los tres tipos de páginas propuestos son lo bastante versátiles como para construir sitios hipermedia útiles.

Utilizando la segunda hipótesis, vemos que uno de los pasos más importantes es la identificación y clasificación de la terminología específica de algún campo de aplicación, una tarea llamada comúnmente Extracción de Terminología [Cabré et al., 2001]. En un texto lineal, las secciones suelen estar claramente marcadas, y las páginas de índice son relativamente fáciles de generar una vez que la terminología es ya conocida. La extracción de terminología es un problema difícil, que ha recibido mucha atención y aún no está totalmente resuelto. Los métodos más corrientes para realizarla suelen suponer que los términos específicos de un campo aparecen con más frecuencia en textos acerca de ese campo que en textos generales. Sin embargo, no siempre ocurre así.

Tras la identificación de los términos relevantes, será útil descubrir —o, al menos, acotar hasta cierto punto— su significado. Hay muchos formalismos con los que se puede codificar la semántica de un lexicón. Algunos de ellos son los utilizados por Sistemas de Representación de Conocimiento como Classic y NeoClassic [Patel-Schneider et al., 1996], o LOOM [MacGregor, 1990]; los que usan formalismos de lógica; o mediante redes semánticas conceptuales. Estas últimas son grafos cuyos nodos representan conceptos y los arcos representan relaciones entre ellos. Es posible obtener inferencias sobre el significado de un concepto si se conoce su posición en la red.

El resultado de esta tarea será el conjunto de secciones que, posiblemente, contendrá el sitio web generado, junto con cierta información acerca de la semántica de los términos identificados, tal como a qué tipo de entidades se refieren (por ejemplo, a personas, lugares, artefactos, etc.)

D.3.2 Generación de contenidos (en línea)

Dado que la hipermedia consta de documentos y enlaces entre ellos, hay dos tipos de métodos que se pueden usar para adaptar un sitio web a diferentes usuarios y dispositivos, como se indicó antes: la presentación

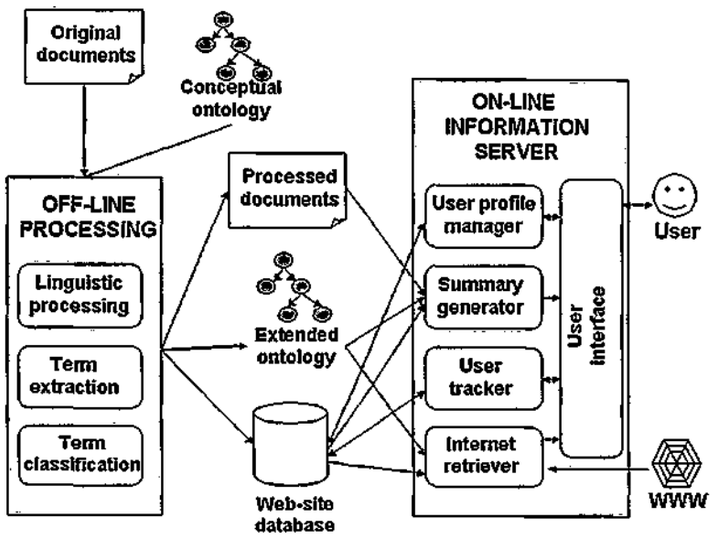


Figure D.1: Diseño de alto nivel de WELKIN.

adaptativa consiste en mostrar diferentes contenidos, por ejemplo, proporcionando la misma información de diferentes maneras, en función de los intereses de los usuarios; y la navegación adaptativa consiste en cambiar la estructura de los enlaces para ayudar a los usuarios a encontrar la información útil.

En lo que respecta a la presentación adaptativa, una vez que se ha decidido cuáles serán las secciones relevantes hay que rellenarlas de contenido. Las páginas de índices puede consistir en listas de términos, y el resto de las páginas se pueden construir combinando fragmentos de los textos originales que sean de interés. Esto incluirá filtrar la información irrelevante, combinar porciones de texto de distintos lugares y, posiblemente, resumir el resultado de acuerdo con los intereses del usuario.

Para este trabajo, a la hora de generar los contenidos de las secciones, se partirá de dos fuentes diferentes: los textos originales proporcionados por el usuario, de los que se tomará la mayor parte de la información, y los resultados de búsquedas automáticas en Internet, para extender la información de los textos originales con datos adicionales.

El soporte a la navegación incluirá las técnicas usadas para enlazar las páginas entre sí.

D.4 La implementación de WELKIN: diseño de alto nivel

Esta sección incluye una descripción de alto nivel de la implementación, llamada WELKIN[2], y una breve descripción de sus componentes. El resto de la tesis describe todos los módulos, con detalles acerca de su implementación y los procedimientos de evaluación.

La Figura D.1 representa el diseño del sistema. La entrada consta de tres recursos: uno o varios documentos, una red semántica conceptual, e Internet (representada al lado derecho de la figura). El procesamiento se puede dividir en dos pasos: el que se realiza fuera de línea, anotando los textos, y el que se realiza en

[2] *Nube, cielo*, en Inglés medieval, proveniente de la palabra de inglés antiguo *wolcen*. Hoy se usa como *cielo* en frases hechas, como *"ring the welkin"*. El nombre de la arquitectura se refiere a las nubes de información que ocultan los datos relevantes y donde hay que buscarlos.

línea, cuando los usuarios acceden al sistema.

D.4.1 Procesamiento fuera de línea

Durante el procesamiento fuera de línea, los textos proporcionados por el usuario son examinados para obtener la información relevante. WELKIN utiliza los siguientes módulos:

Procesamiento lingüístico

Varias herramientas lingüísticas se utilizan para procesar los textos originales. Se describen en detalle en el apéndice B.1, y son las siguientes:

- Un módulo que separa unas palabras de otras.
- Un identificador del comienzo y fin de las oraciones.
- Un analizador morfológico que devuelve el lexema de los nombres y los verbos.
- Un etiquetador que decide a qué parte de la oración pertenece cada palabra (sustantivos, verbos, adverbios, adjetivos, etc.)
- Varios analizadores sintácticos parciales, que identifican cuantificadores, sintagmas nominales y sintagmas verbales.
- Un analizador sintáctico parcial que etiqueta relaciones verbo-sujeto y verbo-complemento directo.
- Un módulo que separa las secciones y los capítulos de los textos originales.

Cada uno de estos módulos añade anotaciones a los textos en forma de entidades y atributos escritos en XML.

Extracción de Terminología

Después del procesamiento lingüístico, hay algunos módulos que localizan varias entidades diferentes. Algunos de estos componentes están especializados en cierto tipo de términos, como fechas o nombres científicos, mientras otros son generales. Utilizan diferentes técnicas, incluyendo expresiones regulares y tecnologías de identificación de idiomas.

Los términos de campos de aplicación particulares que tienen una alta frecuencia de aparición en los documentos se consideran automáticamente términos relevantes, y habrá páginas especiales de hipertexto para describirlos en el sitio web adaptativo generado.

Clasificación de Terminología

Finalmente, hay un módulo que clasifica los términos desconocidos de manera automática en una red semántica conceptual, de manera que se pueda inferir algún significado a partir de su posición en la red.

En resumen, el resultado del procesamiento fuera de línea es el siguiente:

- Los documentos originales se devolverán con anotaciones en XML, acerca de los análisis lingüísticos y la terminología identificada.
- La red semántica será ampliada con los conceptos nuevos.
- Se creará una base de datos sobre estos documentos, que contiene información acerca de cómo ordenar cronológicamente los hechos narrados en el texto; sobre los sitios del documento donde aparecen men-

cionados los términos más relevantes; y se crearán las tablas vacías donde se almacenarán los perfiles de los usuarios del sitio web.

Este entorno permite al autor de los sitios web adaptativos añadir o eliminar módulos especializados en cualquier tipo de terminología, tales como nombres de personas o de empresas.

D.4.2 Procesamiento en línea

Como ya se ha mencionado, el procesamiento en línea tiene lugar cuando los usuarios acceden al sistema para leer la información. Utilizando el gestor de perfiles de usuario, es posible darse de alta en el sistema e inicializar los perfiles. Algunos de los campos del perfil serán obtenidos utilizando pruebas que los nuevos usuarios tienen que realizar.

Después de registrarse, es posible navegar entre las páginas generadas. Si los usuarios tienen restricciones de tiempo, el sistema resumirá los documentos. También es posible pedir al sistema páginas de resúmenes con información obtenida de Internet.

Todos los pasos descritos son completamente automáticos: se puede generar un sitio web completo ejecutando un *script*, sin necesidad de supervisión. Por otra parte, los usuarios experimentados que creen sitios web utilizando WELKIN pueden examinar el resultado de todos y cada uno de los módulos y corregirlos a mano si es necesario.

Gestor de perfiles de usuario

Las páginas web generadas no son complicadas de usar, pero los usuarios han de tener cierta familiaridad con Internet y con los navegadores. Los usuarios nuevos, al registrarse, han de completar unos pocos formularios a partir de los cuales se inicializa su perfil, con los siguientes datos:

1. **La cantidad de información** que desean leer. Se puede indicar de varias maneras: número total de palabras que debe tener el sitio web generado; tasa de compresión que hay que aplicar a todas las páginas; o el tiempo que desea dedicar a leer el sitio web entero. En función de lo que pida el usuario, puede o no ser necesario resumir la información.

2. En el caso de que el usuario indique el tiempo disponible, será preciso conocer su velocidad lectora y su comprensión lectora, medidas en palabras por minuto. Ambas se calculan con una simple prueba.

3. **Intereses**, que pueden escogerse entre una lista de estereotipos predefinidos, o ser generados por el usuario indicando qué textos considera relevantes.

Generador de resúmenes

Cuando un usuario tiene restricciones de tiempo, el sitio web adaptativo debería proporcionar la información de manera condensada, usando algún tipo de resúmenes automáticos. Sin embargo, el usuario debería siempre ser capaz de expandir los resúmenes para leerlos completos, en el caso de que el fragmento se considere interesante y el usuario no quiera perder información sobre él.

Seguimiento del usuario

Aunque los usuarios inicializan su perfil cuando se dan de alta, es posible que un estereotipo se acerque a sus intereses, pero no describa exactamente lo que buscan. En este caso, los usuarios pueden enviar a WELKIN su opinión acerca de si la decisión de seleccionar un párrafo fue correcta o no. El propósito de este componente es actualizar el modelo de intereses de acuerdo con las indicaciones recibidas. Estos cambios tendrán validez inmediata, de manera que la página generada a continuación los reflejará.

Recuperación de datos de Internet

Cuando se identifican los términos importantes de los documentos originales, que darán lugar a secciones separadas en el sitio web, es también posible recuperar desde Internet datos adicionales sobre esos términos. Para realizar la búsqueda en Internet, se utilizan las palabras que aparecen en el contexto de los términos, para aumentar en lo posible la precisión de la información recogida.

Interfaz de usuario

La interfaz de usuario, inicialmente, ha de permitir a los usuarios que se den de alta en el sistema, presentándoles todos los formularios necesarios para inicializar sus perfiles. A continuación, las páginas generadas durante el procesamiento en línea han de incorporar la siguiente funcionalidad:

- Respecto a los *contenidos adaptativos*, se resumirán los textos si es necesario; sin embargo, los usuarios han de poder verlos sin resumir si lo desean. Aunque es de esperar que los documentos originales suministrados por el usuario sean fiables, la información recogida de Internet podría no serlo, por lo que ha de estar claro de dónde proceden los distintos datos. Finalmente, siempre ha de ser posible para el usuario indicar si la información que está recibiendo es interesante o no, para entrenar al sistema.

- Respecto al *soporte de la navegación adaptativo*, se generarán los siguientes enlaces: para navegar por la información en orden cronológico; para leer las descripciones de los conceptos importantes (sobre las personas, lugares o artefactos identificados en los documentos); para leer las secciones linealmente, tal como estaban en los documentos originales; y para leer la información recogida de Internet.

D.5 Recursos de texto utilizados

Esta sección describe los textos originales utilizados para construir algunos sitios web de ejemplo, y los diferentes textos que se han usado, en algún momento de la investigación, para entrenar o probar módulos. Finalmente, para cada módulo hay algunas tablas que indican cuáles son las secciones de la tesis donde se describen, y qué publicaciones hay disponibles acerca de ellos.

D.5.1 Sitios web construidos

Aunque la arquitectura descrita aquí ha sido diseñada de modo general, a la hora de implementarla se realizó con algunas restricciones adicionales. Las siguientes son las restricciones que afectan directamente a los tipos de textos que se pueden utilizar para generar sitios web sobre ellos:

Paso	Módulo	Entrenamiento	Pruebas
Fuera de línea	Procesamiento Lingüístico	Penn Treebank II (WSJ)	WSJ
	Identificación de términos	WN 1.7	LOTR, La Ilíada
	Clasificación de términos	WN 1.7	LOTR, La Ilíada
	- Expresiones temporales		WN 1.7, LOTR, Penn Treebank II
	- Nombres científicos		Los Viajes del Beagle
En línea	Generador de resúmenes		Los Viajes del Beagle, DUC 2003
	Recuperación de Internet		WN 1.7, Los Viajes del Beagle

Table D.1: Corpora utilizados para entrenar o probar los distintos módulos del sistema.

- Los textos procesados deben estar escritos en inglés. Todas las herramientas para el procesamiento lingüístico se han desarrollado para este idioma, de modo que éste es un requisito indispensable que los textos deben satisfacer.

- La identificación y clasificación de términos desconocidos en textos se ha restringido a *entidades físicas*. Éstas incluyen animales, plantas, personas, artefactos, lugares y cuerpos de agua. Los términos que se refieren a abstracciones o acciones no se han estudiado aunque, en teoría, las mismas técnicas podrían clasificarlos en la red semántica. Por tanto, los textos escogidos para generar sitios web adaptativos con WELKIN no deberían contener gran cantidad de abstracciones para que funcione la clasificación de los términos desconocidos.

En segundo lugar, la finalidad del sistema es seleccionar el subconjunto de un texto original que contenga la información más relevante para un usuario particular. Por tanto, para la evaluación, sería deseable procesar textos que puedan ser estudiados desde muchos puntos de vista. De este modo, es posible definir muchos tipos de perfiles de usuario diferentes, y se puede evaluar el grado de adecuación a ellos de la información seleccionada.

Se han desarrollado tres sitios web a partir de textos que satisfacen estos requisitos. Los tres contienen en parte información histórica, aunque desde distintos puntos de vista. Son *Los viajes del Beagle*, de Charles Darwin[3], *La evolución de la Medicina moderna*, de Osler[4], y *Conferencias sobre la historia de la Filosofía* de Hegel[5].

No sólo están escritos o traducidos al inglés, sino que están escritos correctamente, lo que facilita el análisis sintáctico. El hecho de que contengan hechos históricos (los viajes del Beagle, o relatos históricos sobre distintas disciplinas) implica que aparecen términos relevantes que son entidades físicas (personas, lugares, artefactos...). Finalmente, pueden estudiarse desde muchos puntos de vista, dado que contienen fragmentos sobre hechos históricos, descripciones de los lugares donde esos hechos ocurrieron, o descripciones de las subdivisiones de las disciplinas. Como se verá más adelante, la generación de un sitio web completo a partir de textos no lleva mucho tiempo, de modo que se pueden actualizar con cierta frecuencia si el texto original varía con el tiempo (por ejemplo, si se ha generado a partir de noticias de periódico sobre algún tema de interés).

D.5.2 Material de entrenamiento y de pruebas

[3]Obtenido del proyecto Gutenberg, `http://promo.net/pg/`
[4]Obtenido del proyecto Gutenberg
[5]Obtenido de `http://www.class.uidaho.edu/mickelsen/ToC/Hegel-Hist of Phil.htm`

Paso	Módulo	Sección	Publicado como
General	General Architecture		[Alfonseca and Rodríguez, 2002]
Fuera de línea	Procesamiento lingüístico	B.1	[Alfonseca, 2000]
			[Manandhar and Alfonseca, 2000]
	Base de conocimiento	B.2	[Alfonseca, 2002]
	Identificación de terminología	5.4.1, B.2.2	[Alfonseca and Manandhar, 2002a]
	Clasificación de terminología	5 and 6	[Alfonseca and Manandhar, 2002f]
			[Alfonseca and Manandhar, 2002e]
			[Alfonseca and Manandhar, 2002d]
			[Alfonseca and Manandhar, 2002b]
	Expresiones temporales	6.3	[Alfonseca and Manandhar, 2002c]
	Nombres científicos	6.4	
En línea	Perfiles de usuario	8.1	[Alfonseca and Rodríguez, 2003d]
	Generador de resúmenes	8.2	[Alfonseca and Rodríguez, 2003c]
	Interfaz de usuario	8.3	[Alfonseca and Rodríguez, 2003b]
	Recuperación de Internet	8.4	[Alfonseca and Rodríguez, 2003a]

Table D.2: Secciones que describen los módulos principales, y artículos sobre ellos.

Aparte del material ya mencionado, hay otros textos que se han usado para evaluar los diferentes módulos del sistema, mostrados en la tabla D.1:

- El corpus *Penn Treebank II* [Marcus et al., 1993] es una colección de textos anotados con análisis sintácticos. En particular, incluye el corpus del periódico Wall Street Journal (WSJ), una colección de artículos que se ha usado a menudo para entrenar herramientas de procesamiento lingüístico.

- WordNet [Miller, 1995] es una red léxico-semántica en la que las palabras están relacionadas ente sí por medio de relaciones semánticas. Se ha utilizado para entrenar los módulos de *Identificación de Terminología* (para distinguir palabras que representan conceptos generales de las que representan ejemplos particulares de los conceptos) y de *Clasificación de Terminología*. También se ha utilizado como recurso adicional para la identificación de las expresiones temporales y para recoger información nueva de Internet. WordNet será descrito con más detalle en la sección 3.3.

- *El Señor de los Anillos* (LOTR) y *La Ilíada* son dos textos narrativos que se han utilizado para probar la exactitud del módulo de clasificación de terminología cuando se clasifican entidades físicas desconocidas. La elección de dos textos mitológicos se debe al hecho de que incluyen varios términos que se refieren a razas de personas o animales, así como a artefactos y lugares extraños, que pueden usarse para calcular la exactitud del algoritmo de clasificación de entidades físicas.

- La colección de prueba de la *Document Understanding Conference* es un conjunto de textos que se ha utilizado en una competición internacional de resúmenes automáticos.

- Finalmente, *Los Viajes del Beagle* de Darwin se ha utilizado para probar algunos de los módulos del procesamiento en línea y el identificador de nombres científicos.

La tabla D.2 muestra los capítulos y las secciones en las que se describen los módulos más relevantes, y las publicaciones ha que ha dado lugar este trabajo.

D.6 Contribuciones de la tesis

El problema de encontrar información relevante en grandes almacenes de datos y mostrarlo al usuario de manera adaptativa, de manera completamente general, es complicado; esta tesis tiene como finalidad avanzar unos pocos pasos hacia ese objetivo. A la hora de diseñar la arquitectura, se ha puesto especial cuidado en los dos pasos necesarios para generar los sitios web adaptativos: el paso *fuera de línea*, en el que se recoge el conocimiento necesario y se estructuran los contenidos del sitio web; y el paso *en línea*, que incluye las reglas de adaptación a los distintos usuarios.

El paso fuera de línea combina ideas de diferentes campos, tales como Hipermedia Adaptativa, Modelado del Usuario, y varias subáreas del Procesamiento de Lenguaje Natural, como resúmenes automáticos y redes léxico-semánticas. La arquitectura propuesta se ha implementado como una secuencia de módulos, en la que cualquier módulo puede ser intercambiado por otro que escriba el mismo tipo de anotaciones, y los otros componentes no se verán afectados.

Para el paso de identificación y clasificación de terminología, se ha diseñado un nuevo algoritmo que introduce términos nuevos en redes léxico-semánticas, y se ha evaluado con WordNet 1.7 [Miller, 1995], que es una red semántica que contiene 74,487 nodos nominales. En segundo lugar, se ha propuesto un entorno unificado para comparar este algoritmo con posibles alternativas que puedan aparecer. Esto es especialmente relevante, dado que hay pocos algoritmos previos para aprender conceptos nuevos de manera no supervisada, y todos ellos utilizan redes semánticas y textos diferentes, de modo que no es posible compararlos entre sí.

En lo que respecta a la recogida de información de Internet, se ha diseñado un algoritmo nuevo para filtrar los resultados obtenidos con motores de búsqueda convencionales y recoger la información que sea más relevante para los textos originales a partir de los cuales se construyó el sitio web.

Finalmente, en lo que respecta al procesamiento en línea, cuando un usuario nuevo se conecta al sistema, se ha diseñado un nuevo algoritmo de resúmenes automáticos, basado en algoritmos genéticos, para resumir la información de acuerdo con los intereses del usuario.

El capítulo 10 describe con más detalle las contribuciones del trabajo completo y de cada uno de los componentes.

D.7 Estructura de la tesis

Para conseguir el propósito de esta tesis ha sido necesario, como se ha dicho, tratar varios campos diferentes, algunos pertenecientes al procesamiento de lenguaje natural, otros al modelado de usuarios y a la hipermedia adaptativa. Considerando que una descripción del estado del arte de todos estos campos al comienzo de la tesis sería difícil de leer para el lector, las diferentes revisiones de literatura se han separado de acuerdo con el tema de que trata cada parte de la tesis. Por esta razón, las partes II y III incluyen, independientemente, revisiones de literatura que se refieren a sus temas particulares, descripciones del trabajo nuevo desarrollado en esta investigación, y los resultados obtenidos.

La tesis se ha dividido en cuatro partes. La primera contiene este capítulo, que incluye las motivaciones y una descripción de alto nivel de la arquitectura desarrollada para **crear automáticamente sitios web adaptativos**, y la revisión de literatura sobre hipermedia adaptativa. La segunda parte describe el **procesamiento fuera de línea**, durante el cual se genera la base de conocimiento. La tercera parte describe el trabajo realizado para construir los módulos encargados del **procesamiento en línea** y, finalmente, la cuarta parte presenta una evaluación de uso del sistema, junto con las conclusiones.

Cada capítulo comienza con una introducción sobre los puntos que se tratarán y un pequeño párrafo que describe su estructura interna. Además, cada capítulo termina con un breve resumen y, cuando se juzgue necesario, una breve discusión con algunas de las conclusiones preliminares.

Parte I

- El **capítulo 2** revisa el estado del arte de la generación de hipermedia, hipermedia adaptativa y modelado de usuario. Pone especial énfasis en la aplicación de técnicas de Procesamiento de Lenguaje Natural para generación de Hipermedia Adaptativa.

Parte II

El **procesamiento fuera de línea** se centra en la identificación y clasificación de los conceptos relevantes y las secciones del sitio web en los textos originales. Los capítulos incluidos en esta parte son los siguientes:

- El **capítulo 3** contiene una revisión de literatura sobre la adquisición automática de conocimiento léxico: la adquisición de nuevas palabras y sus significados.

- El **capítulo 4** describe la hipótesis de la Semántica Distribucional, que será muy utilizada a lo largo de todo el trabajo. Se discuten varias medidas de distancia entre los significados de los conceptos, y se presenta una justificación empírica de la hipótesis.

- El **capítulo 5** describe las técnicas desarrolladas en esta investigación para extender automáticamente redes léxico-semánticas con conceptos nuevos tomados de textos. Esto incluye los pasos de Identificación de Terminología y de Clasificación de Terminología.

- El **capítulo 6** describe otras técnicas que se han utilizado, fuera del ámbito de la Semántica Distribucional, para la clasificación de terminología. Principalmente se utilizan expresiones regulares y otros tipos de patrones.

Parte III

El **procesamiento en línea** incluye los módulos que interaccionan con el visitante del sitio web generado. La parte III describe los componentes del sistema que, usando la información generada en la parte II y la información disponible sobre el usuario, construyen sobre la marcha las páginas hipermedia y los enlaces a otras páginas.

- El **capítulo 7** describe el estado del arte en resúmenes automáticos.

- El **capítulo 8** describe los módulos que generan las páginas de hipertexto de acuerdo con los intereses del usuario, y crean sobre la marcha los hiperenlaces que estructuran el sitio web generado. Este capítulo describe el modelo de usuario utilizado y cómo afecta a la presentación de contenidos. Describe un nuevo algoritmo para producir resúmenes automáticos. Finalmente, describe también los módulos que recogen información de Internet para completar las páginas hipermedia con datos adicionales.

Parte IV

- **El capítulo 9** contiene la evaluación del sistema. Describe tres ejemplos de sitios web generados, y un experimento controlado con usuarios.
- **El capítulo 10** contiene las conclusiones del trabajo: las contribuciones, una comparación con el trabajo previo en esta área, y las líneas abiertas para el trabajo futuro, tanto desde el punto de vista general de la arquitectura como para cada uno de los módulos separadamente.

Apéndices

Finalmente, varios apéndices contienen información adicional relevante, aunque no central, para los propósitos de esta tesis:

- El **apéndice A** contiene una lista de abreviaturas.
- El **apéndice B** describe el trabajo adicional que fue necesario realizar para este trabajo, como las herramientas lingüísticas de proceso de textos, el diseño de las bases de datos, etc.
- El **apéndice C** muestra varios ejemplos de textos generados por diferentes módulos para la adaptación de contenidos: la selección de información relevante para el usuario, el resultado de los resúmenes automáticos, y la información recogida automáticamente de Internet.
- Los **apéndices D y E** son la traducción al español de los capítulos de introducción y conclusiones.

Appendix E

Conclusiones y trabajo futuro

El trabajo descrito en esta tesis presenta varias contribuciones al campo de Hipermedia Adaptativa. Además, describe varios avances en algunas de las técnicas colaterales que se han utilizado, y que incluyen las ontologías léxicas, los resúmenes automáticos, y las búsquedas en Internet, entre otras. Este capítulo describe las contribuciones del trabajo y el posible trabajo futuro.

E.1 Contribuciones

Integración de diversas técnicas

Un aspecto importante de este trabajo es la manera como se han integrado diferentes algoritmos, componentes y técnicas para alcanzar el objetivo. Se han aplicado técnicas de Procesamiento de Lenguaje Natural para la adquisición de conocimiento léxico y generación de textos; se han estudiado problemas formulados por la iniciativa de la web semántica con respecto a la recolección de datos relevantes en Internet; y se han aplicado varios métodos y técnicas de hipermedia adaptativa, con los datos generados por los módulos previos, a la hora de generar las páginas de hipertexto. La arquitectura integra más de quince componentes diferentes escritos en lenguajes de programación distintos (C, C++, flex, prolog, Java, *scripts* de Linux y Javascript), que se intercambian información a través de sus interfaces o por medio de anotaciones XML, de modo que todas las anotaciones generadas por algún módulo serán usadas, de una u otra manera, por alguno de los demás.

Una nueva arquitectura

Se ha descrito una nueva arquitectura para transformar automáticamente textos lineales en sitios hipermedia adaptativos. Incorpora técnicas de procesamiento de textos para analizar los textos lineales y extraer información. El entendimiento de textos se realiza de manera parcial, infiriendo el significado de los conceptos nuevos clasificándolos en WordNet. Al combinar técnicas de generación de hipermedia con modelado del usuario, el texto proporcionado por el sistema está adaptado a los intereses del usuario. Se han diseñado nuevas maneras de representar los intereses del usuario, para permitir pequeñas variaciones en la definición de estos intereses.

Este sistema se ha implementado y probado. Se ha aplicado a distintos textos y temas, con buenos resultados. También se ha evaluado en un experimento controlado, que fue bien recibido por los usuarios.

El experimento sugiere que la productividad aumenta cuando se realizan ciertas tareas de análisis, y los usuarios consideraron el sistema fácil de usar y atractivo.

La construcción automática de sitios hipermedia se puede realizar entre una hora y hora y media, sin necesidad de supervisión. Si se han de definir estereotipos para el sitio web, eso puede llevar entre una o dos horas más para entrenar al sistema, dado que los textos han de ser anotados a mano. Esto quiere decir que un sitio hipermedia completo puede obtenerse rápidamente a partir de texto lineal, y en unos pocos días podría estar lista una colección de sitios de hipermedia adaptativa. Comparado con el trabajo de crear un sitio web a partir de textos electrónicos manualmente, utilizando editores de texto o de hipertexto, mejora mucho la productividad, aparte de ofrecer los resultados de la clasificación de terminología, las facilidades de adaptación y de generación de resúmenes, y las diferentes opciones de navegación.

La arquitectura modular permite que un módulo pueda ser fácilmente reemplazado por otro, sin alterar las porciones restantes. De hecho, algunos de los módulos, como el componente de clasificación de terminología o algunos de los algoritmos de procesamiento lingüístico, fueron sustituidos sobre la marcha conforme iban siendo mejorados sin alterar el funcionamiento del sistema. Esto facilitará, en el futuro, la incorporación de posibles mejoras en los componentes, tales como nuevos módulos para reconocer terminología especializada, o las descritas en la sección E.4 más adelante.

Búsquedas precisas en Internet

Se ha realizado un módulo opcional que realiza búsquedas en Internet y permite que el usuario recupere información adicional muy precisa sobre los conceptos relevantes que aparecen en la web adaptativa. Esta fue una de las características más apreciadas por los usuarios durante la evaluación, y ofrece muchas posibilidades. Si alguien necesita recuperar información relevante sobre conceptos de dominios específicos, y hay uno o varios documentos sobre ellos, es posible usar este módulo para recoger la información y realizar las búsquedas. De este modo, puede utilizarse independientemente del resto del sistema como programa de recuperación de información especializado, ejecutándose por encima de los motores de búsqueda tradicionales.

Componentes

Este trabajo también describe los algoritmos nuevos para cada uno de los componentes del sistema, algunos de los cuales tienen aplicación para muchos otros sistemas o arquitecturas, aparte de la generación de hipermedia. Cada uno de estos componentes se ha evaluado independientemente, y se ha observado que su precisión es similar a la de otras técnicas del estado del arte. Para uno de ellos, el componente de clasificación de terminología, se ha realizado un conjunto de pruebas que se ha hecho público.

La sección E.4 contiene una discusión separada sobre cada uno de los componentes, y describe las ideas para posibles mejoras en cada uno de ellos.

E.2 Comparación con otros trabajos

El trabajo descrito en esta tesis combina ideas de modelado de usuario, hipermedia, conversión de texto lineal en hipertexto y procesamiento de lenguaje natural. Por tanto, cada uno de los diferentes algoritmos y decisiones de diseño, que pueden estudiarse separadamente, se ha comparado a lo largo de la tesis con el trabajo relacionado en cada una de esas áreas restringidas. Esta sección, por el contrario, se centrará en

otros trabajos existentes que se dirigen al objetivo principal: la generación automática de sitios hipermedia a partir de texto.

Blustein [1994] distingue tres tipos de sistemas de conversión de texto en hipertexto. En primer lugar están los que crean un sitio web a partir de un **texto muy estructurado**, como definiciones de diccionarios [Raymond and Tompa, 1988]; o los sistemas que esperan una serie de **anotaciones** acerca de la estructura del texto lineal, a partir de las cuales se pueda obtener fácilmente la estructura del sitio web [Furua et al., 1989]. Con estos sistemas, la estructura del texto o las anotaciones se usarán para dividir el texto en hiperpáginas y para añadir los enlaces. Esta información puede consistir en indicaciones acerca de dónde comienzan o terminan las secciones del documento, o referencias a otras porciones del documento (que pueden convertirse en enlaces), escritas en lenguajes como *roff o LaTeX.

Estos sistemas de generación de texto a hipertexto suelen comenzar dividiendo el texto en porciones más pequeñas. Las anotaciones contenidas en el documento original indican dónde comienza y termina cada porción. Aunque no haya anotaciones, si el formato del texto es conocido, también es posible segmentarlo automáticamente, identificando títulos de secciones y cambios de párrafo. En el caso particular de diccionarios o enciclopedias en hipertexto, generalmente hay abreviaturas y marcas que indican la función de las distintas partes de las definiciones (etiquetas morfológicas, acepciones, referencias a otras palabras, etc).

WELKIN tiene en común con estos sistemas que es completamente automático: lo único que necesita son los textos lineales originales, y un *script* realiza todo el procesamiento sin ninguna supervisión por parte del usuario. WELKIN, además, proporciona las siguientes ventajas:

- El procesamiento lingüístico de los textos permite adquirir cierto conocimiento a partir de ellos, como la clasificación de las palabras que aparecen con mayor frecuencia.
- No son necesarias las anotaciones en el texto, tales como mandatos de LaTeXo SGML.
- El resultado final se genera dinámicamente para cada tipo de usuario. Por otra parte, siempre es posible leer los textos sin ninguna adaptación, si el usuario indica que está interesado en todo y que no quiere que se resuman los textos. En este caso, tendrá toda la información original disponible.

Hay otros métodos para la segmentación de textos que no requieren anotaciones [Hearst, 1993]. Estos métodos buscan *unidades temáticas*, es decir, porciones del texto que se refieren al mismo tema. WELKIN combina ambos procedimientos, ya que

- Los cambios de sección y de párrafo se identifican automáticamente estudiando la estructura del documento, como su indentación, frases subrayadas o que comienzan con numeración, frases en mayúsculas, y palabras clave como *prólogo*, *capítulo* o *apéndice*.
- Las unidades temáticas se identifican durante el paso de filtrado, a partir del perfil de intereses del usuario. Además, es posible clasificar una porción de texto como perteneciente a varios intereses a la vez, algo que ocurre a menudo en los textos, pues temas diferentes pueden solaparse.

El segundo método para la generación de hipermedia utiliza técnicas de Inteligencia Artificial para transformar texto en hipertexto. Dos de los sistemas que primero utilizaron esta técnica son VISAR [Clitherow et al., 1989] y TOPIC [Hahn and Reimer, 1988]. El primero se utilizó para mantener una base de datos de citas de artículos de periódico, y el segundo representa la estructura temática de un texto como un grafo jerárquico, estableciendo relaciones entre los fragmentos del texto. Ambos utilizan procesamiento lingüístico

hasta cierto punto, y sólo son válidos en áreas restringidas de conocimiento, porque utilizan bases de datos de conocimiento específico y no son fáciles de adaptar a distintos tipos de textos.

Comparado con estos sistemas, WELKIN tiene en común que también realiza un cierto procesamiento lingüístico de los textos y utiliza una base de conocimiento léxico (WordNet). Sin embargo, la diferencia principal es que WordNet es una base de conocimiento de propósito general, y el conocimiento específico (los términos desconocidos) se adquieren automáticamente, lo que facilita el trabajo con textos sobre temas distintos, como biología, historia, medicina o filosofía. Además, es capaz de procesar textos arbitrarios recogidos de Internet.

Finalmente, hay un tercer tipo de sistemas que se utiliza cuando se parte de varios documentos distintos. En este caso, se pueden utilizar técnicas de Recuperación de Información para agrupar los documentos que estén relacionados entre sí, y añadir enlaces entre los que tienen más relación. [Blustein, 1994] indica que se puede utilizar técnicas como el modelo vectorial para calcular similitudes ente textos. Esta es la técnica utilizada por WELKIN para realizar el filtrado temático de los párrafos, donde cada párrafo se representa con el vector de las palabras que contiene, para calcular su similitud con el modelo de intereses del usuario.

En resumen, las tres técnicas diferentes se han aplicado a problemas diferentes durante el diseño de WELKIN. Como puede comprobarse, se han integrado en un único sistema, aplicando cada una de ellas al aspecto en el que es más útil.

En general, hay pocos trabajos que estudien el tema de transformar textos lineales en hipermedia adaptativa. Los siguientes párrafos describen las diferencias entre la arquitectura descrita en esta tesis y otros trabajos relacionados.

Algunos sistemas utilizan técnicas de generación automática de textos para producir hipermedia. Oberlander et al. [1998] y Milosavljevic et al. [1998] describen dos sistemas para crear guías virtuales para museos y enciclopedias adaptativas, y ambos tienen la ventaja de que utilizan un componente de generación de lenguaje natural. De esta forma, el texto generado presenta mucha mayor flexibilidad para adaptarse a los usuarios, dado que las mismas frases pueden adaptarse completamente a sus intereses y conocimientos previos.

Por otra parte, utilizar un sistema de generación de lenguaje natural tiene el problema de que la creación de la base de conocimiento suele ser muy trabajosa. Oberlander et al., y Milosavljevic et al., utilizaron técnicas semiautomáticas para adquirir la información a partir de textos **semi-estructurados**. Por ejemplo, Milosavljevic et al. [1998] utilizó una base de datos de un museo, en la cual parte de la información, como la fecha o el lugar de procedencia de un objeto, estaba almacenada en campos diferentes de una tabla; además, intentó algunas técnicas simples de extracción de información en campos que contenían texto no restringido. WELKIN, dado que utiliza una técnica de resúmenes en vez de una generación completa de lenguaje, pierde cierta flexibilidad, pero por otra parte genera sitios web completos a partir únicamente de textos escritos en lenguaje natural.

Lo mismo podría aplicarse con respecto al trabajo descrito por DiMarco et al. [1997], que describe un sistema para proporcionar consejos médicos. El resultado que se muestra a los usuarios se produce con técnicas de generación de lenguaje, utilizando *documentos maestros* que contienen la información que se mostrará a los usuarios con perfiles diferentes. Para adquirir estos documentos maestros se utilizan muchas técnicas de procesamiento de lenguaje, tales como un analizador sintáctico semi-automático, un programa para identificar correferencias, y un analizador retórico, pero finalmente los documentos son revisados por un "escritor técnico profesional o diseñador de documentos web". Aunque la generación de lenguaje natural permite adaptar el resultado al usuario de forma mucho más sofisticada, el sistema de utilizar resúmenes

adaptados libera al diseñador del sitio web del problema de adquisición de conocimiento.

Para finalizar, Ragetli [2001] describe un procedimiento para estructurar un conjunto de páginas dándoles estructura de hipermedia, creando primero una jerarquía de conceptos y asociando después los documentos a los conceptos. Sin embargo, hay varias diferencias entre ese sistema y WELKIN:

- La jerarquía se construye manualmente a partir de los términos obtenidos de un glosario, en un proceso que lleva mucho trabajo. Se describen algunos métodos que podrían ayudar en el proceso, pero no se evalúan. Por ejemplo, Ragetli señala que algunos de los conceptos se clasifican utilizando emparejamiento de palabras [Neville-Manning et al., 1999]; de esta forma, se puede descubrir la relación entre *transporte aéreo* y *transporte*, o entre *probabilidad condicional* y *probabilidad*. Es una buena idea, pero aún debe ser refinada, pues no explica cómo distinguir términos complejos como *probabilidad condicional* de palabras que ocurren juntas sin formar un término, como *aceptación condicional*. Las técnicas de identificación de terminología serían de gran ayuda en este problema [Vivaldi, 2002].
- Los documentos se asocian a los conceptos utilizando técnicas de recuperación de información basadas en el modelo vectorial.

Hay varias diferencias entre este trabajo y WELKIN, unas pocas de las cuales son las siguientes:

- WELKIN analiza la estructura interna de los documentos, y genera las páginas a partir de porciones de los documentos originales. Algunas de las páginas generadas podrían ser una combinación de fragmentos obtenidos de documentos diferentes, o de fragmentos no consecutivos de un mismo documento. En el sistema descrito por Ragetli, cada página hipermedia es un documento original completo.
- En WELKIN, hay un modelo de usuario, y los contenidos mostrados al usuario dependen de su perfil. Incluso la estructura del sitio web es dinámica.
- Finalmente, Ragetli [2001] describe un trabajo en fase de diseño, que aún ha de ser implementado y probado.

E.3 Trabajo futuro

Siguen algunas ideas acerca de cómo se podría mejorar este trabajo, o aplicarlo a problemas diferentes:

- Adquiriendo información, no sólo de textos electrónicos, sino también de hipertexto estático existente, sería posible crear un sistema que produjese un sitio hipermedia adaptativo a partir de una combinación de textos y sitios web existentes. Esto puede ser útil para generar vistas personalizadas de intranets o de sitios web, mostrando a cada usuario tan sólo la información más relevante y omitiendo el resto. Todos los enlaces a páginas juzgadas irrelevantes serían eliminados, de manera que el hiperespacio sería reducido a sólo los datos de interés.
- En línea con el comentario previo, otra mejora útil sería la adición de un motor de búsqueda que mire en todas las secciones relevantes del sitio web adaptativo. Con la arquitectura actual no es una tarea trivial, considerando que las páginas web se generan sobre la marcha, de modo que sería necesario producirlas todas e indexarlas cada vez que los intereses del usuario cambien. Una posible solución sería realizar la búsqueda sobre los textos originales, y a continuación detectar cuáles son las páginas hipermedia que contienen cada una de las porciones relevantes y evaluar tan sólo esas páginas en cuanto a su relevancia para el usuario.

- También sería útil crear algún sistema que exporte los resultados a otros formatos. Por ejemplo, alguien podría estar interesado en crear un sitio web estático sobre *Los Viajes del Beagle* desde el punto de vista de la geografía. En este caso, WELKIN podría generar el sitio completo, y almacenarlo en HTML.

- La evaluación de uso se ha realizado mediante un experimento controlado, pero sería muy interesante repetirla con un conjunto mayor de usuarios que necesiten utilizar el sistema para su trabajo o por cualquier otra razón, de manera que dediquen más tiempo al sistema. Si este experimento se realizara en el futuro, se podrían analizar los archivos que registran las acciones realizadas, para extraer más conclusiones acerca de las acciones más frecuentes realizadas por los usuarios, los puntos débiles del sistema, las acciones que son más lentas de ejecutar, etc.

E.4 Discusión y trabajo futuro de los componentes

La arquitectura descrita se divide en varios componentes, cada uno de los cuales se puede implementar internamente de muchas maneras posibles. Las ventajas y limitaciones de las distintas técnicas utilizadas en los módulos se han descrito ya previamente, en las secciones que detallan su evaluación. Esta sección enumera los componentes más relevantes, junto con ideas acerca de cómo podrían mejorarse. Algunas de las líneas abiertas son lo bastante complejas como para convertirse en proyectos de investigación centrados tan sólo en ellas.

E.4.1 Clasificación de terminología

El trabajo presentado aquí para la clasificación de terminología es original en el sentido en que es, por lo que yo sé, el primer método completamente no supervisado para extender ontologías léxicas con nuevos términos. Las técnicas previas, o bien necesitan heurísticas, o bien requieren que un juez valide los resultados, o incluyen reglas y *scripts* que tienen que ser codificadas antes de usar el sistema. Con el nuevo método presentado aquí, tan sólo es necesaria la ontología, el documento que contiene los nuevos términos, y una conexión a Internet para realizar las búsquedas. Todo el procesamiento se realiza automáticamente.

La mayor parte de las limitaciones del componente actual han sido ya descritas en la tesis. Los puntos flacos más importantes son:

- Tan sólo se proporciona un hiperónimo para cada término desconocido. Sería deseable, por ejemplo, saber que *Chiloé* es tanto una isla como un distrito administrativo.

- La clasificación depende de los contextos en los que aparezcan las palabras, y por tanto es necesario ver la palabra en contexto cierto número de veces para que la clasificación sea precisa. Los términos que aparecen con poca frecuencia no serán clasificados correctamente con esta técnica.

- Algunas distinciones de detalle no son fáciles de realizar a partir de los contextos. Términos que son semánticamente muy distintos, tales como un lugar, una persona o un río, sin duda aparecerán en contextos muy distintos; sin embargo, las palabras que están muy relacionadas aparecerán prácticamente en los mismos contextos, y un corpus pequeño podría no reflejar las diferencias. Este es el caso de la distinción entre varones y mujeres, que resultó especialmente difícil de realizar en los experimentos.

Varias mejoras que pueden realizarse son las siguientes:

- El algoritmo podría implementarse como un sistema de búsqueda en haz, de manera que se exploren varios caminos al mismo tiempo. Esta modificación sería deseable por dos razones: primero, si el algoritmo realizara cualquier decisión incorrecta, sería imposible corregirla, porque sólo se avanza hacia abajo en la ontología. Utilizando la búsqueda en haz, al final se ordenarían los resultados por orden de similitud, para asociar el término nuevo a uno o más hiperónimos.

- Otra línea abierta para investigación futura consiste en adquirir contextos para todos los synsets de WordNet, incluyendo las entidades físicas restantes y otros tipos de conceptos, como características psicológicas, acciones, o grupos. Esto no se ha hecho hasta ahora por las limitaciones en la velocidad de la red y del procesador y los requisitos de almacenamiento, pero es de esperar que conforme mejoren los sistemas informáticos será posible recogerlas todas.

- El diseño de WordNet ha sido criticado con frecuencia, aunque hay que tener en cuenta que incluso las teorías psicológicas en las que se sustentaron decisiones de diseño han variado a lo largo de estos años. Algunas de las críticas más comunes son que hay menos enlaces de hiperonimia de los que podría haber, y que se distinguen demasiados sentidos de algunas palabras. Podría valer la pena comprobar si nuestro método funciona con otras ontologías, como la de alto nivel de Cyc [Lenat, 1995].

- La precisión en la clasificación podría mejorar usando otras características, además de los contextos en que aparecen los términos. Por ejemplo, si se analiza la tabla 9.1, se observa que varios términos que comienzan con la palabra *Mr.* fueron clasificados como mujeres. Simples heurísticas tales como el conocimiento de que algunos títulos personales sólo pueden aplicarse al sexo masculino podrían conducir a una mejor clasificación de estos synsets. De la misma manera, se podría extender WordNet con otras características para bloquear algunos caminos de búsqueda en el caso de que el género, el número u otra restricción no concuerden con el concepto nuevo.

- En cuanto a la representación de los contextos, se podrían examinar otras relaciones sintácticas. Además, se podrían generalizar de manera adecuada las palabras en la *topic signature*, para aumentar la precisión en la clasificación. Algunas maneras posibles de realizar la generalización son los procedimientos descritos por Hearst and Schutze [1993] o por Hovy and Lin [1999].

- Agirre et al. [2001] indica que algunos filtros sobre los documentos recuperados de Internet, tales como escoger sólo un documento de cada sitio web, podrían mejorar la calidad.

- Finalmente, es bastante complicado el problema de descubrir si una palabra se está usando con dos significados en un texto dado (por ejemplo, *St. Jago* puede referirse a una persona y a un lugar en el mismo documento). Una solución parcial podría consistir en utilizar técnicas como reconocimiento de entidades, tomadas del campo de Extracción de Información, para realizar una primera clasificación de las palabras.

  Igualmente, queda como línea futura el descubrir si un término que ya está presente en WordNet se está usando o no con el mismo significado en un texto dado.

Como comentario final, este tipo de procesamiento también podría aplicarse para agrupar conceptos por medio de *clustering* (utilizando la medida de similitud entre los contextos), para crear una ontología a partir de cero; o para estudiar las maneras en que cambia el uso de una palabra a lo largo de la historia, por citar unos pocos ejemplos.

E.4.2 Semántica Distribucional

En cuanto a la hipótesis de la semántica distribucional, aunque ha sido aplicada a muchos problemas, hay pocos trabajos que intenten demostrar, ya sea teóricamente o empíricamente, su validez. Uno de estos trabajos es el libro de Levin [1993], un estudio bastante completo sobre las propiedades distribucionales de los verbos ingleses.

La sección 4.3 describe una manera en la cual se puede comprobar que existe correlación entre la similitud semántica de varios conceptos de WordNet y la similitud de los contextos en que aparecen. Sin embargo, sería de interés generalizar este trabajo seleccionando aleatoriamente conjuntos mayores de synsets de WordNet aleatoriamente, y comprobando con tests estadísticos que la correlación se aplica para cualesquiera dos synsets seleccionados al azar.

Podría ser el caso de que la correlación fuese mayor para unos fragmentos de WordNet que para otros. Es de esperar, dados los resultados del algoritmo de clasificación de terminología, que algunas distinciones semánticas, como las diferencias entre hombres y mujeres, sean más difíciles de capturar con los contextos, de modo que es de esperar que se podrán extraer conclusiones interesantes de estos experimentos. Finalmente, los mismos experimentos se pueden realizar con palabras que se encuentren en el mismo synset, para comprobar si son realmente sinónimas o plesiónimas.

Es importante notar que, para estos experimentos, ha de tenerse especial cuidado en que los documentos recogidos de Internet contengan las palabras con su significado correcto. Podría ser útil utilizar las técnicas de filtrado descritas en la sección previa para la recolección de los datos de Internet, y explorar nuevas maneras de asegurar que las palabras se utilizan con los sentidos correctos.

En cuanto a las *signatures*, es posible que una generalización de las palabras con conceptos de WordNet mejore los resultados de las clasificaciones.

E.4.3 Análisis de expresiones temporales

En cuanto a la identificación e interpretación de las expresiones temporales en los textos, es posible realizar algunas mejoras:

- En primer lugar, la evaluación se ha realizado con conjuntos de prueba pequeños, y sería deseable repetirla con textos mayores. Setzer and Gaizauskas [2001] describen un estudio en el que diferentes anotadores humanos señalaron los eventos y las relaciones temporales encontradas en unos textos, pero hubo poco acuerdo entre ellos, un hecho que indica que sería necesario definir más claramente las guías para la anotación. Mientras tales recursos no estén disponibles, será difícil evaluar estos sistemas.
- También es importante identificar la correferencia de los eventos, de modo que dos referencias al mismo evento sean reconocidas.
- Algunas partes del sistema, tales como la desambiguación del significado de las palabras, están en el origen de muchos de los errores del análisis de las expresiones temporales, por lo que deberían ser mejorados.
- Las edades de las personas pueden también considerarse indicadores de tiempo.
- Habría que extender el sistema para representar intervalos de tiempo que no estén claramente especificados. Por ejemplo, la expresión *hace un año* puede referirse a un instante indeterminado dentro de un período de tiempo bastante indeterminado.

E.4.4 Generador de resúmenes

Este trabajo incluye un nuevo algoritmo de generación automática de resúmenes basado en extracción de frases con algoritmos genéticos. Este método es muy sencillo de programar, por lo que sería muy fácil de transportar a otros entornos o lenguajes de programación.

Tal como se indicó en la sección 7.2.1, la mayor parte de los procedimientos para generar resúmenes utilizando técnicas de extracción tienen una desventaja importante en el hecho de que no tienen en cuenta la tasa de compresión a la hora de seleccionar las oraciones. Con el ejemplo ya usado anteriormente, si las dos oraciones con mayor puntación, s_1 y s_2, proporcionan conjuntamente una idea, y la tercera oración s_3 proporciona ella sola otra idea de interés, entonces un resumen de una oración debería seleccionar s_3, que contiene una idea completa, antes que s_1 o s_2 por separado.

Este hecho es especialmente relevante para resúmenes multidocumento. Si las dos oraciones de mayor peso, provenientes de diferentes documentos, proporcionan la misma idea, sólo se debería seleccionar una de ellas para el extracto. Con el método de los algoritmos genéticos, se evalúan los extractos completos, no las oraciones separadamente, por lo que la función de ajuste podría adaptarse en esta línea.

Otra característica que se puede incorporar con facilidad es la habilidad de generar resúmenes adaptados a los usuarios o a puntos de vista particulares. Teniendo en cuenta, en la función de ajuste, la similitud entre cada oración y el perfil de intereses del usuario, es posible adaptar los resúmenes generados.

En resumen, con el algoritmo genético, las decisiones más importantes recaen ahora en la función de ajuste. Como consecuencia de esto, la mayor parte de las decisiones de trabajo futuro tienen que ver con esta función de ajuste, que puede tener en cuenta muchos más factores, como medidas de cohesión de los extractos, solapamiento entre las oraciones seleccionadas, etc.

Los algoritmos genéticos tienen la ventaja añadida de que la búsqueda se realiza de manera no lineal, y pueden utilizarse en aplicaciones prácticas. El algoritmo utilizado es bastante básico, pero existen muchas posibilidades de mejorarlo, tales como realizar el cruzamiento de los cromosomas de formas más sofisticadas, o comenzar con resúmenes pequeños y hacerlos crecer hasta la longitud deseada. Utilizar algoritmos de búsqueda local después de que el algoritmo genético haya terminado, es otra posibilidad que también suele dar buenos resultados.

La función de ajuste propuesta es experimental. Es posible que la combinación lineal de los diferentes valores no sea la mejor solución, de manera que se han de explorar otras posibilidades. Aunque el sistema aquí presentado no es supervisado, se podría hacer que la misma función de ajuste evolucionara junto con los resúmenes, en un entorno supervisado, para ajustar mejor los pesos.

Nótese que, en realidad, el algoritmo no restringe la longitud de los resúmenes. Se puede definir fácilmente el genotipo como un vector booleano con ceros y unos, y establecer que las oraciones extraídas son aquellas tales que su gen correspondiente tiene un valor 1. De este modo, se puede trabajar con resúmenes de longitud variable.

Finalmente, el algoritmo de resúmenes debería extenderse con algún tipo de procesamiento lingüístico, como resolver los antecedentes de los pronombres, de manera que un pronombre cuyo antecedente haya sido eliminado pueda ser reemplazado por él. El procesamiento lingüístico también resultaría de utilidad para juntar las oraciones del resumen resultante de modo que se obtenga un resultado más condensado.

Este algoritmo ha sido utilizado para participar en la competición de resúmenes automáticos *Document Understanding Conference* del año 2003.

E.4.5 Modelado de usuario

El componente de modelado de usuario permite que los usuarios especifiquen sus intereses y el grado de compresión que se aplicará a los textos. Hay varias maneras de mejorarlo:

- Actualmente, hay dos modos de definir los perfiles de usuario personalizados (diferentes de los estereotipos). La primera, que consiste en anotar a mano un centenar de párrafos para entrenar al clasificador, es un proceso bastante largo. La otra opción disponible consiste en declarar interés por todo y, después, mientras se lee el sitio web, ir descartando párrafos que no sean de interés. Sería interesante investigar otros procedimientos para indicar los intereses del usuario.

- Modelando los conocimientos del usuario, es posible no repetir información, y el sistema puede aprovechar esos datos para añadir explicaciones adicionales, comparando los temas actuales con otra información que el usuario ya ha visitado.

- Se podrían incluir en el modelo de usuario otras características, como la edad o el idioma. Esto, combinado con un procedimiento de generación de lenguaje, permitiría presentar los textos con distintos estilos de lenguaje.

- En principio, el modelo de usuario podría extenderse con cualquier otro tipo de información, como características psicológicas, dispositivos utilizados, etc.

- El diseño actual de las páginas web adaptativas, con tres marcos en la ventana del navegador, no es apropiado para verlo con bajas resoluciones de pantalla, ya que es necesario utilizar las barras de desplazamiento para ver todos los contenidos de las páginas. Además, algunos usuarios han sugerido que la utilización de colores más claros sería más agradable a la vista.

E.4.6 Recogida de documentos de Internet

Con respecto a la recogida de información adicional de Internet, la sección 8.4 describe una manera nueva de aumentar la precisión de la información recogida, comparándola con los contextos en que aparecen los términos en los documentos originales. Las heurísticas y los filtros que se han utilizado son nuevos y ayudaron a eliminar la mayor parte de los datos irrelevantes.

Los resúmenes generados aún se pueden mejorar. Las siguientes son algunas ideas para aumentar su calidad:

- Reordenar los párrafos dentro del documento de resumen de acuerdo con el tema que tratan. Algunos de los párrafos recogidos se refieren a la historia de una ciudad, mientras que otros se refieren a su economía, su población y folklore, etc. Los párrafos que tratan del mismo tema deberían aparecer contiguos en el resumen.

- Eliminar los párrafos que contienen la misma información. De hecho, muchos de los resúmenes generados contienen un párrafo repetido varias veces, dado que aparecía en más de un documento. Algunas veces, aparecen párrafos enteros con varias copias casi idénticas.

  Un caso más complejo se da cuando varios párrafos contienen solapamientos. En este caso, sería deseable juntar la información que proporcionan en un sólo párrafo, utilizando herramientas de comprensión y generación de lenguaje natural.

- Identificación de contradicciones. En algunos de los resúmenes generados había párrafos que proporcionaban información contradictoria, tales como fechas diferentes para los mismos eventos. Sería deseable poder identificarlos y proporcionar la fuente de la que se obtuvieron todos los materiales, de modo que el usuario juzgue cuál es la más creíble.

Finalmente, este procedimiento puede extenderse con los servicios proporcionados por `Google` y otros buscadores de buscar imágenes en Internet, de modo que se puedan recoger imágenes relevantes a los términos. Adicionalmente, con técnicas de procesamiento de imágenes, sería posible determinar si una imagen es una fotografía o un diagrama (por ejemplo, un mapa), de modo que sea factible clasificarlas y mostrarlas dependiendo de los intereses del usuario.

References

E. Agirre, O. Ansa, E. Hovy, and D. Martinez. Enriching very large ontologies using the www. In *Ontology Learning Workshop, ECAI*, Berlin, Germany, 2000a.

E. Agirre, O. Ansa, D. Martínez, and E. Hovy. Enriching wordnet concepts with topic signatures. In *Proceedings of the NAACL workshop on WordNet and Other lexical Resources: Applications, Extensions and Customizations*, Pittsburg, 2001.

E. Agirre, X. Arregi, X. Artola, A. Díaz de Ilarraza, K. Sarasola, and A. Soroa. A methodology for building translator-oriented dictionary systems. *Machine Translation Journal*, 2000b.

E. Agirre and G. Rigau. Word sense disambiguation using conceptual density. In *Proceedings of the COLING'96*, 1996.

E. Alfonseca. *NP-chunking with transformation lists.* Trabajo tutelado número 73126 para la obtención del grado de suficiencia investigadora. Departamento de Ingeniería Informática, Universidad Autónoma de Madrid, 2000.

E. Alfonseca. A wordnet interface to apl2. In *APL-2002 Conference. Also published as E. Alfonseca and S. Manandhar, A WordNet Interface to APL2, APL Quote Quad (ACM SIGAPL), Vol. 32:4, p. 7-16*, Madrid, Spain, 2002.

E. Alfonseca, M. DeBoni, J. L. Jara-Valencia, and S. Manandhar. A prototype question-answering system using syntactic and semantic information for answer retrieval. In *Text REtrieval Conference, TREC-2001, Question-Answering Track*, 2001.

E. Alfonseca and S. Manandhar. Distinguishing instances and concepts in wordnet. In *Poceedings of the First International Conference on General WordNet*, Mysore, India, 2002a.

E. Alfonseca and S. Manandhar. Extending a lexical ontology by a combination of distributional semantics signatures. In *Proceedings of EKAW'02*, Siguenza, Spain, 2002b. Also published in *Knowledge Engineering and Knowledge Management.* Lecture Notes in Artificial Intelligence 2473. Springer Verlag.

E. Alfonseca and S. Manandhar. A framework for constructing temporal models from texts. In *Florida Artificial Intelligence Research Society conference, FLAIRS-2002*, Pensacola, Florida, U.S.A., 2002c.

E. Alfonseca and S. Manandhar. Improving an ontology refinement method with hyponymy patterns. In *Language Resources and Evaluation (LREC-2002)*, Las Palmas, 2002d.

E. Alfonseca and S. Manandhar. Proposal for evaluating ontology refinement methods. In *Language Resources and Evaluation (LREC-2002)*, Las Palmas, 2002e.

E. Alfonseca and S. Manandhar. An unsupervised method for general named entity recognition and automated concept discovery. In *Poceedings of the First International Conference on General WordNet*, Mysore, India, 2002f.

E. Alfonseca and P. Rodríguez. Automatically generating hypermedia documents depending on user goals. In *Workshop on Document Compression and Synthesis in Adaptive Hypermedia Systems, AH-2002*, Malaga, Spain, 2002.

E. Alfonseca and P. Rodríguez. Acquisition of domain-dependent summaries from the www, 2003a. Submitted.

E. Alfonseca and P. Rodríguez. An adaptive user interface in hypermedia sites generated from linear text, 2003b. Submitted.

E. Alfonseca and P. Rodríguez. Generating extracts with genetic algorithms, 2003c. to appear in European Conference on Information Retrieval (ECIR-2003).

E. Alfonseca and P. Rodríguez. Modelling users' interests and needs for an adaptive on-line information system, 2003d. to appear in Proceedings of UM-2003.

R. Alterman. A dictionary based on concept coherence. *Artificial Intelligence*, 25:153–186, 1985.

R. Alterman and L. A. Bookman. Reasoning about a semantic memory encoding of the connectivity of events. *Cognitive Science*, 16:205–232, 1992.

H. Assadi. *Construction d'ontologies á partir de textes techniques.* Ph. D. thesis, L'Université Paris 6, 1998.

J. Atserias, L. Carmona, I. Castellón, and S. Cervell. Morphosyntactic analysis and parsing of unrestricted spanish text. In *Proceedings of the 1th International Conference on Language Resources and Evaluation*, Granada, Spain., 1998.

R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval.* Addison-Wesley-Longman, Harlow, UK, 1999.

J. Baldridge, T. Morton, and G. Bierner. Quipu maxent, https://sourceforge.net/projects/maxent/, 2001.

R. Basili, R. Catizone, M. T. Pazienzia, M. Stevenson, P. Velardi, M. Vindigni, and Y. Wilks. An empirical approach to lexical tuning. In *Proceedings of the Workshop Adapting Lexical and Corpus Resources to Sublanguages and Applications, LREC First International Conference on Language Resources and Evaluation*, Granada, Spain, 1998.

I. H. Beaumont. User modelling in the interactive anatomy tutoring system anatom-tutor. *User Modelling and User-Adapted Interaction*, 4(1):21–45, 1994. Reprinted in Adaptive Hypertext and Hypermedia, pp. 91-116. Kluwer Academic Publishers, 1998.

A. Bell. *The Discourse Structure of News Stories. In (Allan Bell and Peter Garret, eds.), Approaches to Media Discourse*, chapter 3, pages 64–104. Blackwell Publishers, 1998.

S. D. Benford, I. Taylor, D. Brailsford, B. Koleva, M. Craven, M. Fraser, G. T. Reynard, and C. M. Greenhalgh. Three-dimensional visualisation of the world wide web. In *ACM Computing Surveys Symposium on Hypertext and Hypermedia*, 1999.

A. L. Berger, S. A. Della Pietra, and V. J. Della Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71, 1996. URL `citeseer.nj.nec.com/berger96maximum.html`.

T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, 284(5):34–43, may 2001.

D. M. Bikel, R. Schwartz, and R. M. Weischedel. An algorithm that learns what's in a name. *Machine Learning*, 34:211, 1999.

W. J. Blustein. *An Evaluation of Tools for Converting Text to Hypertext.* Ms.C. Thesis, Department of Computer Science, The University of Western Ontario, Canada, 1994.

A. Borgida. Description logics and predicate logics. *Artifical Intelligence Journal*, 82:353–367, 1996.

A. Borthwick, J. Sterling, E. Agichtein, and R. Grishman. Exploiting diverse knowledge sources via maximum entropy in named entity recognition. In *Proceedings of the Sixth Workshop on Very Large Corpora*, New Brunswick, New Jersey, 1998.

P. Bouillon. *The adjective* vieux*: The point of view of "Generative Lexicons". In Evelyne Viegas (ed.), Breadth and Depth of Semantic Lexicons*, pages 147–166. Kluwer Academic Publishers, 1999.

C. Boyle and A. O. Encarnacion. Metadoc: an adaptive hypertext reading system. *User modeling and user-adapted interaction*, 4:1–19, 1994.

T. Brants. *TnT - A Statistical Part-of-Speech Tagger.* User manual, 2000.

W. F. Brewer. *The story schema: universal and culture-specific properties In David R. Olson, Nancy Torrance and Angela Hildyard (eds.) Literacy, Language and Learning: the Nature and Consequences of Reading and Writing*, pages 167–94. Cambridge University Press, 1985.

E. Brill. Transformation-based error-driven learning and natural language processing: A case study in part of speech tagging. *Computational Linguistics*, 21(4):543–565, 1995.

T. Briscoe and J. Carroll. Automatic extraction of subcategorization from corpora. In *Proceedings of the 5th Conference on Applied Natural Language Processing (ANLP-97)*, Washington DC, USA, 1997.

P. Brusilovsky. Methods and techniques of adaptive hypermedia. *User Modeling and User Adapted Interaction*, 6(2-3):87–129, 1996.

P. Brusilovsky. Adaptive hypermedia. *User Modelling and User-Adaptive Interaction, Ten Year Anniversary Issue (Alfred Kobsa, ed.)*, 11(1/2):87–110, 2001.

P. Brusilovsky, A. Kovsa, and J. Vassileva. *Adaptive Hypertext and Hypermedia.* Kluwer Academic Publishers, Dordrecht, 1998.

P. Brusilovsly, J. Eklund, and E. Schwarz. Web-based education for all: a tool for development of adaptive courseware. In *Proceedings of the Seventh International World Wide Conference. Also published as Computer Networks and ISDN Systems, 30(1-7), pp. 291–300*, 1998.

P. Buitelaar. CoreLex: *Systematic Polysemy and Underspecification.* Ph.D. Thesis. Brandeis University, Department of Computer Science, 1998.

S. Busemann and H. Horacek. A flexible shallow approach to text generation. In *Proceedings of the 9th International Workshop on Natural Language Generation, INLG'98*, pages 238–247, 1998.

M. T. Cabré, R. Estopá, and J. Vivaldi. *Automatic term detection: a review of current systems. In Recent advances in computational terminology, volume 2 of Natural Language Processing*, pages 53–87. John Benjamins, 2001.

J. Carbonell, D. Harman, E. Hovy, S. Maiorano, J. Prange, and K. Sparck-Jones. Vision statement to guide research in question & answering and text summarisation., 2001. URL `http://www-nlpir.nist.gov/projects/duc/roadmapping.html`.

G. Carenini, V. Mittal, and J. D. Moore. Geneating patient specific interactive natural language explanations. In *Proceedings of the Eighteenth Symposium on Computer Applications in Medical Care*, Banff, Canada, 1994.

G. Carenini, M. Ponzi, and O. Stock. Combining natural language and hypermedia as new means for information access. In *Proceedings of the 5th European conference of cognitive ergonomics, ECCE'90*, Urbino, Italy, 1990.

J. Carletta. Assessing agreement on classification tasks: the kappa statstic. *Computational Linguistics*, 22 (2):249–254, 1996.

R. Carro. *Un mecanismo basado en tareas y reglas para la creación de sistemas hipermedia adaptativos: aplicación a la educación a través de Internet.* Ph.D. Thesis, Universidad Autónoma de Madrid, Spain, 2001.

R. Carro, E. Pulido, and P. Rodríguez. *Developing and accessing adaptive internet-based courses*, chapter 4, pages 111–152. World Scientific, 2002.

R. M. Carro, E. Pulido, and P. Rodríguez. Designing adaptive web-based courses with tangow. In *Proceedings of the 7th International Conference on Computers in Education, ICCE'99*, pages 697–704, Chiba, Japan, 1999.

E. Carter. *Quantitative analysis of hypertext generation and organisation techniques.* Ph.D. thesis, department of Artificial Intelligence, University of Edinburgh, 1996.

C. A. Carver, R. A. Howard, and E. Lavelle. Enhancing student learning by incorporating student learning styles into adaptive hypermedia. In *Proceedings of ED-MEDIA'96 - World Conference on Educational Multimedia and Hypermedia*, pages 118–123, Boston, MA, 1996.

W. B. Cavnar and J. M. Trenkle. N-gram-based text categorization. In *Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*, pages 161–175, Las Vegas, U.S., 1994. URL `citeseer.nj.nec.com/68861.html`.

D. Chakrabarti, D. K. Narayan, P. Pandey, and P. Bhattacharyya. Experiences in building the indo wordnet: A wordnet for hindi. In *Poceedings of the First International Conference on General WordNet*, Mysore, India, january 2002.

K. Church, W. Gale, P. Hanks, and D. Hindle. *Using Statistics in Lexical Analysis. In U. Zernik (ed.), Lexical Acquisition: Exploiting On-line Resources to Build a Lexicon*, chapter 6, pages 115–164. Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1991.

P. Clark and B. W. Porter. Building concept representations from reusable components. In *AAAI/IAAI*, pages 369–376, 1997.

P. Clark, J. Thompson, and B. Porter. A knowledge-based approach to question-answering. In *R. Fikes and V. Chaudhri, editors, Proc. AAAI'99 Fall Symposium on Question-Answering Systems. AAAI*, 1999. URL `citeseer.nj.nec.com/128575.html`.

P. Clitherow, D. Riecken, and M. Muller. Visar: A system for inference and navigation in hypertext. In *Proceedings of Hypertext'89*, pages 293–304, 1989.

P. Cohen, V. Chaudhri, A. Pease, and R. Schrag. Does prior knowledge facilitate the development of knowledge-based systems. In *Proceedings of AAAI-99*, 1999.

A. Copestake. The acquilex lkb: representation issues in semi-automatic acquisition of large lexicons. In *3rd Conference on Applied Natural Language Processing (ANLP-92)*, Trento, Italy, 1992.

A. Copestake, J. Carroll, R. Malouf, and S. Oepen. The (new) lkb system, user manual, 2000.

M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, Kamal Nigam, and Seán Slattery. Learning to construct knowledge bases from the world wide web. *Artificial Intelligence*, 118(1–2):69–113, 1999.

D. P. da Silva, R. van Durm, E. Duval, and H. Olivi. Concepts and documents for adaptive educational hypermedia: a model and a prototype. In *Proceedings of the Second Workshop on Adaptive Hypertext and Hypermedia at the Ninth ACM Conference on Hypertext and Hypermedia*, pages 35–43, Pittsburth, USA, 1998.

I. Dagan, A. Itai, and U. Schwall. Two languages are more informative than one. In *proceedings of ACL-91*, pages 130–137, Berkeley, California, 1991.

J. Daudé, L. Padró, and G. Rigau. Mapping wordnets using structural information. In *38th Anual Meeting of the Association for Computational Linguistics (ACL'2000)*, Hong Kong, 2000.

P. de Bra, A. Aerts, D. Smits, and N. Stash. Aha! meets aham. In *Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 381–384. Springer-Verlag, 2002. Also published as LNCS 2347.

P. de Bra, P. Brusilovsky, and G.-J. Houben. Adaptive hypermedia: From systems to framework. *ACL Computing Surveys*, 31(4), 1999a.

P. de Bra and L. Calvi. AHA! an open adaptive hypermedia architecture. *The New Review of Hypermedia and Multimedia*, 4:115–139, 1998.

P. de Bra, G.-J. Houben, and H. Wu. AHAM: A dexter-based reference model for adaptive hypermedia. In *UK Conference on Hypertext*, pages 147–156, 1999b. URL `citeseer.nj.nec.com/debra99aham.html`.

B. de Carolis, F. de Rosis, C. Andreoli, V. Cavallo, and M. L. de Cicco. The dynamic generation of hypertext presentations of medical guidelines. *The New Review of Hypermedia and Multimedia*, 4:67–88, 1998.

W. Degen, B. Heller, H. Herre, and B. Smith. Gol: Towards an axiomatized upper-level ontology. In *Proceedings of the International Conference on Formal Ontology in Information Systems, FOIS-2001*, 2001.

C. DiMarco, G. Hirst, and E. Hovy. Generation by selection and repair as a method for adapting text for the individual reader. In *Proceedings of the Flexible Hypertext Workshop, 8th ACM International Hypertext Conference*, Southampton, UK, 1997.

J. Domingue, M. Martins, J. Tan, A. Stutt, and H. Pertusson. Alice: Assisting online shoppers through ontologies and novel interface metaphors. In *Proceedings of EKAW'02*, Siguenza, Spain, 2002. Also published in *Knowledge Engineering and Knowledge Management.* Lecture Notes in Artificial Intelligence 2473. Springer Verlag.

K. Dunn and R. Dunn. *Teaching students through their individual learning styles.* National Council of Principles, Reston, VA, 1978.

K. D. Eason. Towards the experimental study of usability. *Behaviour and Information Technology*, 3(2): 133–143, 1993.

P. Edmonds and G. Hirst. Near synonymy and lexical choice. *Computational Linguistics*, 2002.

H. P. Edmundson. New methods in automatic abstracting. *Journal of the Association for Computational Machinery*, 16(2):264–286, 1969.

F. Esposito, S. Ferilli, N. Fanizzi, and G. Smeraro. Learning from parsed sentences with inthelex. In *Proceedings of CoNLL-2000 and LLL-2000*, pages 198–198, Lisbon, Portugal, 2000.

A. Farquhar, R. Fikes, and J. Rice. *Tools for assembling modular ontologies in ontolingua.* AAAI Press, Menlo Park, California, 1997.

D. Faure and C. Nédellec. A corpus-based conceptual clustering method for verb frames and ontology acquisition. In *LREC workshop on Adapting lexical and corpus resources to sublanguages and applications*, Granada, Spain, 1998.

E. Filatova and E. Hovy. Assigning time-stamps to event-clauses. In *Workshop on Temporal and Spatial Information Processing, ACL-2001*, Toulouse, France, 2001.

J. Fink, A. Kobsa, and A. Nill. Adaptable and adaptive information provision for all users, including disabled and elderly people. *The New Review of Hypermedia and Multimedia*, 4:163–188, 1998.

J. Firth. *Papers in Linguistics 1934-1951.* Oxford University Press, London, 1957a.

J. Firth. *A synopsys of linguistic theory 1930-1955. In F. Palmer (ed.), Selected Papers of J. R. Firth.* Longman, London, 1957b.

M. S. Fox and M. Gruninger. Ontologies for enterprise integration. In *Conference on Cooperative Information Systems*, pages 82–89, Univ. Toronto, 1994. URL `citeseer.nj.nec.com/fox94ontologie.html`.

J. M. Fritz. *An Investigation of the Effectiveness of Open Hypertext Techniques for Qualitative Decision Support.* Ph.D. Dissertation. University of York, U.K., 1995.

R. Furua, C. Plaisant, and B. Shneiderman. A spectrum of automatic hypertext constructions. *Hypermedia*, 1(2):179–195, 1989.

R. Gaizauskas, T. Wakao, K. Humphreys, H. Cunningham, and Y. Wilks. University of sheffield: Description of the lasie system as used for muc-6. In *Proceedings of the Sixth Message Understanding Conference (MUC-6)*, pages 207–220. Morgan Kauffmann, 1995.

S. Geldof. Con-textual navigation support. *The new Review of multimedia and hypermedia*, 4:47–66, 1998.

S. Geldof. From context to sentence form. In *Elhadad, M. (ed.) Proceedings of the International Conference on Natural Language Generation, INLG'2000*, pages 225–231, Mitzpe Ramon, Israel, 2000. New Brunswick, NJ: ACL.

J. E. Gilbert and C. Y. Han. Adapting instructions in search of 'a significant difference'. *Journal of Network and Computer Applications*, 22, 1999.

J. Gonzalo, F. Verdejo, I. Chugur, and J. Cigarran. Indexing with wordnet synsets can improve text retrieval: In *Proceedings of the COLING/ACL '98 Workshop on Usage of WordNet for NLP*, pages 38–44, Montreal, Canada, 1998. URL `citeseer.nj.nec.com/gonzalo98indexing.html`.

G. Grefenstette. Automatic thesaurus generation from raw text using knowledge-poor techniques. In *Making Sense of Words. Ninth Annual Conference of the UW Centre for the New OED and text Research*, 1993.

T. R. Gruber. A translation approach to portable ontologies. *Knowledge Acquisition*, 5(2):199–220, 1993.

U. Hahn. Topic parsing: Accounting for text macro structures in full-text analysis. *Information Processing and Management*, 26(1):135–170, 1990.

U. Hahn and U. Reimer. Automatic generation of hypertext knowledge bases. In *Proceedings of the ACM conference on office information systems*, pages 182–188, 1988.

U. Hahn and K. Schnattinger. Towards text knowledge engineering. In *AAAI/IAAI*, pages 524–531, 1998. URL `citeseer.nj.nec.com/43410.html`.

M. A. K. Halliday. *Language as Social Semiotic.* Edward Arnold, London, 1978.

M. A. K. Halliday and R. Hasan. *Cohesion in Text.* Longmans, London, 1996.

S. Harabagiu and S: Maiorano. Multi-document summarization with GISTEXTER. In *Proceedings of the Third Language Resources and Evaluation Conference (LREC-2002)*, Las Palmas, 2002.

D. Harman. *(ed.) NIST Special Publication 500-207: The First Text REtrieval Conference (TREC-1).* NIST, 1991. URL `http://trec.nist.gov/pubs/trec1/t1_proceedings.html`.

Z. Harris. *Mathematical structures of Language.* Wiley, New York, 1968.

P. M. Hastings. *Automatic acquisition of word meaning from context.* University of Michigan, Ph. D. Dissertation, 1994.

F. Hayes-Roth, D. A. Waterman, and D. B. Lenat (eds.). *Building Expert Systems.* Addison-Wesley, Reading, Massachusetts, 1983.

M. A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of COLING-92,* Nantes, France, 1992.

M. A. Hearst. Texttiling: A quantitative approach to discourse segmentation, 1993.

M. A. Hearst. *Automated Discovery of WordNet Relations. In Christiane Fellbaum (Ed.) WordNet: An Electronic Lexical Database,* pages 132–152. MIT Press, 1998.

M. A. Hearst and H. Schutze. Customizing a lexicon to better suit a computational task. In *Proceedings of the ACL SIGLEX Workshop,* Columbus, Ohio, 1993.

D. Hindle. Noun classification from predicate-argument structures. In *Proceedings of the 28th Annual Meeting of the Association of Computational Linguistics,* pages 268–275, Pittsburgh, 1990.

T. Hirashima, N. Matsuda, T. Nomoto, and J. Toyoda. Context-sensitive filtering for browsing in hypertext. In *Proceedings of International Conference on Intellingent User Interfaces, IUI'98,* pages 21–28, San Francisco, U.S.A., 1998.

J. Hobbs. *On the Coherence and Structure of Discourse.* Report No. CSLI-85-37. Stanford, California: Center for the Study of Language and Information, Stanford University, 1985.

J. Hockenmaier, G. Bierner, and J. Baldridge. Providing robustness for a ccg system. In *Proceedings of the Workshop on Linguistic Theory and Grammar Implementation,* Hong Kong, 2000.

J. Hockenmaier and M. Steedman. Generative models for statistical parsing with combinatory categorial grammar. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics,* Philadelphia, 2002.

J. Holland. *Adaptation In Natural and Artificial Systems.* University of Michigan Press, 1975.

Homer. The iliad.

J. Hothi and W. Hall. An evaluation of adapted hypermedia techniques using static user modelling. In *Proceedings of the Second Adaptive Hypertext and Hypermedia Workshop at the Ninth ACM International Hypertext Conference, Hypertext'98,* pages 45–50, Pittsburgh, PA, 1998. Computer Science Reports 98/12, Eindhoven University of Technology.

E. Hovy and C-Y. Lin. *Automated Text Summarization in SUMMARIST,* pages 81–94. I. Mani and M. T. Maybury (eds.) Advances in Automatic Text Summarization. MIT Press, Cambridge, Massachusetts, 1999.

D. Hull and G. Grefenstette. Querying across languages: A dictionary-based approach to multilingual information retrieval. In *Proceedings of ACM/SIGIR,* 1996.

S. Huttunen, R. Yangarber, and R. Grishman. Diversity of scenarios in information extraction. In *Proceedings of 3rd International Conference on Language Resources and Evaluation: LREC-2002*, Las Palmas de Gran Canaria, Spain, 2002.

N. Ide. Corpus encoding standard (ces) - version 1.5., 2000. URL `http://www.cs.vassar.edu/CES`.

N. Ide and J. Véronis. Introduction to the special issue on word sense disambiguation: the state of the art. *Computational Linguistics*, 24:1–40, 1998.

J. W. Irwin. The effect of linguistic cohesion on prose comprehension. *Journal of Reading Behaviour*, 12(4): 325–332, 1980.

M. D. Jackson and J. L. McClelland. Processing determinants of reading speed. *Journal of experimental psychology*, 108:151–181, 1979.

P. Jacobs and L. Rau. Scisor: Extracting information from on-line news. *Communications of the ACM*, 33 (11):88–97, 1990.

A. Jameson. *What can the rest of us learn from research on adaptive hypermedia –and vice-versa*, chapter preface. P. Brusilovsky and A. Kovsa and J. Vassileva (eds.) Adaptive Hypertext and Hypermedia. Kluwer Academic Publishers, Dordrecht, 1998.

J. S. Justeson and S. M. Katz. Technical terminology: some linguistic properties and an algorithm for identification in text. *Natural Language Engineering*, 1(1):9–27, 1995.

M.-Y. Kan, K. R. McKeown, and J. L. Klavans. Domain-specific informative and indicative summarization for information retrieval. In *Proceedings of Document Undestanding Conference, DUC-2001*, 2001.

J. Kietz, A. Maedche, and R. Volz. A method for semi-automatic ontology acquisition from a corporate intranet. In *Workshop "Ontologies and text", co-located with EKAW'2000*, Juan-les-Pins, French Riviera, 2000.

A. Kilgarriff. Evaluationf word sense disambiguation programs: Progress report. In *Proceedings of the SALT Workshop on Evaluation in Speech and Language Technology*, Sheffield, 1997.

A. Kobsa, J. Koenemann, and W. Pohl. Personalized hypermedia presentation techniques for improving on-line customer relationships, 1999.

D. B. Koen and W. Bender. Time frames: Temporal augmentation of the news. *IBM Systems Journal*, 39 (3/4), 2000.

S. Landes, C. Leacock, and R. Tengi. *Building Semantic Concordances. In (C. Fellbaum (Ed.) WordNet: An Electronic Lexical Database*, chapter 8, pages 199–216. MIT Press, 1998.

M. Laroussi and M. Benahmed. Providing an adaptive learning through the web case of cameleon: Computer aided medium for learning on networks. In *Proceedings of CALISCE'98, 4th Int. Conf. on Computer Aided Learning and Instruction in Science and Engineering*, pages 411–416, Goteborg, Sweden, 1998.

C. Lee, G. Lee, and S. J. Yun. Automatic wordnet mapping using word sense disambiguation. In *38th Anual Meeting of the Association for Computational Linguistics (ACL'2000)*, Hong Kong, 2000.

L. Lee. *Similarity-Based Approaches to Natural Language Processing.* Ph.D. thesis. Harvard University Technical Report TR-11-97, 1997.

D. Lenat. *Steps to Sharing Knowledge.* Mars N., editor, Towards Very Large Knowledge Bases. IOS Press, 1995.

D. B. Lenat and R. V. Guha. *Building Large Knowledge-Based Systems.* Addison-Wesley, Reading (MA), USA, 1990.

A. Lenci, R. Bartolini, N. Calzolari, A. Agua, S. Busemann, E. Cartier, K. Chevreau, and J. Coch. Multilingual summarisation by integrating linguistic resources in the mlis-musi project. In *Proceedings of the Third Language Resources and Evaluation Conference (LREC-2002)*, Las Palmas, 2002.

B. Levin. *English Verb Classes and Alternations: A Preliminary Investigation.* University of Chicago Press, Chicago, IL., 1993.

H. Li and N. Abe. Generalising case frames using a thesaurus and the mdl principle. *Computational Linguistics*, 24(2):217–244, 1997.

C.-Y. Lin. *Robust Automated Topic Identification.* Ph.D. Thesis. University of Southern California, 1997.

C-Y. Lin and E. Hovy. Identifying topics by position. In *Proceedings of the 5th Applied Natural Language Processing Conference*, pages 283–290, New Brunswick, New Jersey, 1997.

C.-Y. Lin and E. Hovy. Neats: A multidocument summarizer. In *Proceedings of Document Undestanding Conference, DUC-2001*, 2001.

R. Longacre. *The paragraph as a grammatical unit. In T. Givon(ed.), Syntax and Semantics. Volume 12, Discourse and Syntax*, volume 12, pages 115–134. Academic Press, New York, 1979.

J. López-Cuadrado, T. A. Pérez, R. Arruabarrena, J. A. Vadillo, and J. Gutiérrez. Generation of computerized adaptative tests in an adaptative hypermedia system. In *International Conference on Information and Comunication Technologies in Education (ICTE 2002)*, Badajoz (Spain), 2002.

J. Lyons. *A structural theory of semantics and its applications to lexical sub-systems in the vocabulary of Plato.* Ph. D. thesis, University of Cambridge, England. Published as *Structural Semantics*, No. 20 of the Publications of the Philological Society, Oxford, 1963, 1961.

S. Lytinen. A unification-based, integrated natural language processing system. *Computers and Mathematics with Applications*, 23(6-9):403–418, 1991.

M. J. Maña-López, M. de Buenaga-Rodríguez, and J. M. Gómez Hidalgo. Diseño y evaluación de un generador de resúmenes de texto con modelado de usuario en un entorno de recuperación de información. In *XIV Congreso de la Sociedad Española de Procesamiento de Lenguaje Natural (SEPLN'98)*, Alicante, Spain, 1998. Published in *Procesamiento del Lenguaje Natural*, 23, september 1998, 32–39.

R. MacGregor. *LOOM User's Manual.* ISI/WP-22. USC/Information Sciences Institute, 1990.

C. Macleod and R. Grishman. Comlex syntax reference manual, 1994.

A. Maedche and S. Staab. Ontology learning for the semantic web. *IEEE Intelligent systems*, 16(2), 2001.

B. Magnini and M. Speranza. Merging global and specialized linguistic ontologies. In *3rd International Conference on Language Resources and EvaluationConference (LREC) and OntoLex2002 workshop*, Las Palmas, 2002.

B. Magnini and C. Strapparava. Improving user modeling with content-based techniques. In *in M. Bauer, P.J. Gmytrasiewicz and J. Vassileva (eds), User Modeling 2001: 8th International Conference.* Springer-Verlag, Berlin Heidelberg, 2001.

S. Manandhar and E. Alfonseca. Noun phrase chunking with apl2. In *Proceedings of the APL-Berlin-2000 Conference, Berlin. Also published as E. Alfonseca and S. Manandhar, Noun Phrase chunking with APL2, APL Quote Quad (ACM SIGAPL), Vol. 30:4, p. 135-143*, 2000.

I. Mani. *Automatic Summarization.* John Benjamins Publishing Company, 2001.

I. Mani and E. Bloedorn. Machine learning of generic and user-focused summarization. In *Proceedings of the Fifteenth National Conference of Artificial Intelligence (AAAI'98)*, pages 821–826, Menlo Park, California, 1998.

I. Mani and E. Bloedorn. Summarising similarities and differences among related documents. *Information Retrieval*, 1(1):35–67, 1999.

I. Mani, T. Firmin, D. House, M. Chrzanowski, G. Klein, L. Hirschman, B. Sundheim, and L. Obrst. *The TIPSTER SUMMAC Text Summarization Evaluation: Final Report.* MITRE Technical Report MTR 98W0000138. McLean, VA: The MITRE Corporation, 1998.

I. Mani and G. Wilson. Robust tempral processing of news. In *38th Anual Meeting of the ACL (ACL'2000)*, Hong Kong, 2000.

W. C. Mann and S. A. Thompson. Rhetorical structure theory: Towards a functional theory of text organisation. *Text*, 8(3):243–281, 1988.

D. Marcu. *The rhetorical parsing, summarization and generation of natural language texts.* Ph. D. Thesis, University of Toronto, Canada, 1997.

D. Marcu. *Discourse Trees are good indicators of importance in text. In I. Mani and M. T. Maybury (eds.), Advances in Automatic Text Summarisation*, pages 123–136. MIT Press, Cambridge, Massachusetts, 1999.

D. Marcu. Discourse-based sumarization in duc-2001. In *Proceedings of Document Undestanding Conference, DUC-2001*, 2001.

D. Marcu and L. Gerber. An inquiry into the nature of multidocument abstract. In *Proceedings of the NAACL'01 workshop on text summarisation*, Pittburgh, PA, 2001.

M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz. Building a large annotated corpus of english: the penn treebank. *Computational Linguistics*, 19(2):313–330, 1993.

B. Masand, G. Linoff, and D. Waltz. Classifying news stories using memory based reasoning. In *Proceedings of SIGIR 92*, pages 59–65, 1992.

M. T. Maybury. Generating summaries from event data. *Information Processing and Management*, 31(5): 733–751, 1995. Reprinted in *Advances in Automatic Text Summarization*, I. Mani and M. T. Maybury (eds.), 165–281. Cambridge, Massachusetts: MIT Press.

K. F. McCoy. Highlighting a user model to respond to misconceptions. In *A. Kobsa and W. Wahlster (eds.): User Models in Dialog Systems*, pages 233–254. Springer-Verlag, 1989.

T. McKinley. Managing all information assets. *Document Management Magazine*, July/August 1997.

C. Mellish, M. O'Donnell, J. Oberlander, and A. Knott. An architecture for oportunistic text generation. In *Proceedings of the 9th International Workshop on Natural Language Generation, INLG'98*, pages 28–37, 1998.

A. Mikheev. Periods, capitalized words, etc. *Computational Linguistics*, 28(3):245–288, 2002.

A. Mikheev, C. Grover, and M. Moens. Description of the ltg system used for muc-7. In *Proceedings of 7th Message Understanding Conference (MUC-7)*, 1998.

G. A. Miller. WordNet: A lexical database for English. *Communications of the ACM*, 38(11):39–41, 1995.

G. A. Miller. *Nouns in WordNet*, pages 23–46. Christiane Fellbaum (Ed.) WordNet: An Electronic Lexical Database. MIT Press, 1998.

M. Milosavljevic. Augmenting the user's knowledge via comparison. In *A. Jameson, C. Paris and C. Tasso (eds.), Proceedings of the 6th International Conference on User Modeling, UM97*, pages 119–130. Wien: SpringerWienNewYork, 1998.

M. Milosavljevic, R. Dale, S. J. Green, C. Paris, and S. Williams. Virtual museums on the information superhighway: Prospects and potholes. In *Proceedings of CIDOC'98, the Annual Conference of the International Committee for Documentation of the International Council of Museums*, Melbourne, Australia, 1998.

S. Mohanty, N. B. Ray, R. C. B. Ray, and P. K. Santi. Oriya wordnet. In *Poceedings of the First International Conference on General WordNet*, Mysore, India, january 2002.

T. Móia. Telling apart temporal locating adverbials and time-denoting expressions. In *Workshop on Temporal and Spatial Information Processing, ACL-2001*, Toulouse, France, 2001.

R. Montague. *Formal Philosophy.* New Haven: Yale University Press, 1974.

MUC6. *Proceedings of the 6^{th} Message Understanding Conference (MUC-6).* Morgan Kaufman, 1995.

MUC7. *Proceedings of the 7^{th} Message Understanding Conference (MUC-7).* Morgan Kaufman, 1998.

S. Mukherjea. Information visualization for hypermedia systems. In *ACM Computing Surveys Symposium on Hypertext and Hypermedia*, 1999.

T. Murray, T. Shen, J. Piemonte, C. Condit, and J. Thibedeau. Adaptivity in the metalinks hyper-book authoring framework. In *Proceedings of the International Workshop on Adaptive and Intelligent Web-based Educational Systems, ITS'2000*, Osnabruck: Technical Report of the Institute for Semantic Information Processing, 2000.

R. Navigli and P. Velardi. Automatic adaptation of wordnet to domains. In *3rd International Conference on Language Resources and EvaluationConference (LREC) and OntoLex2002 workshop*, Las Palmas, 2002.

C. G. Neville-Manning, I. H. Witten, and G. W. Paynter. Lexically-generated subject hierarchies for browsing large collections. *International Journal of Digital Libraries*, 2(3):111–123, 1999.

M. Ng, W. Hall, P. Maier, and R. Armstrong. History-based link annotation for self-exploratory learning in web-based hypermedia, 2001. URL `citeseer.nj.nec.com/489055.html`.

J. Nielsen. *Usability Engineering*. Academic Press, Boston, Mass., 1990.

S. Nirenburg, S. Beale, and K. Mahesh. Lexicons in the mikrokosmos project. In *Proceedings of the AISB'96 Workshop on Multilinguality in the Lexicon*, Brighton, UK, 1996.

T. Nomoto. Moddbs-x^m: A diversity based summarizer for duc2001. In *Proceedings of Document Understanding Conference, DUC-2001*, 2001.

E. Not, D. Petrelli, M. Sarini, O. Stock, C. Strapparava, and M. Zancanaro. Hypernavigation in the physical space: adapting presentation to the user and to the situational context. *New Review of Multimedia and Hypermedia*, 7(4):223–237, 1998.

E. Not and M. Zancanaro. Content adaptation for audio-based hypertexts in physical environments. In *Proceedings of the 2nd Workshop on Adaptive Hypertext and Hypermedia, held in conjunction with the Ninth ACM Conference on Hypertext and Hypermedia, Hypertext '98*, pages 25–31, Pittsburgh, PA, USA, 1998.

J. Oberlander, M. O'Donell, C. Mellish, and A. Knott. Conversation in the museum: experiments in dynamic hypermedia with the intelligent labeling explorer. *The new review of multimedia and hypermedia*, 4:11–32, 1998.

D. O'Sullivan, A. McElligott, and R. F. E. Sutcliffe. Augmenting the princeton wordnet with a domain specific ontology. In *Proceedings of the Workshop on Basic Issues in Knowledge Sharing at the 14th International Joint Conference on Artificial Intelligence*, Montreal, 1995.

P. Paredes and P. Rodríguez. Considering sensing-intuitive dimension to exposition-exemplification in adaptive sequencing. In *Proceedings of AH'2002*, Málaga, Spain, 2002.

C. L. Paris. The use of explicit user models in a generation system for tailoring answers to the user's level of expertise. In *A. Kobsa and W. Wahlster (eds.): User Models in Dialog Systems*, pages 200–232. Springer-Verlag, 1989.

P. F. Patel-Schneider, M. Abrahams, L. A. Resnick, D. L. McGuinness, and A. Borgida. *NeoClassic Reference Manual: Version 1.0*. Artificial Intelligence Principles Research Department, AT&T Labs Research, 1996.

D. Pelleg and A. Moore. X-means: Extending k-means with efficient estimation of the number of clusters. In *Proceedings of the Seventeenth International Conference on Machine Learning, ICML2000*, pages 727–734, Stanford University, CA, 2000. Morgan Kauffman.

G. Petasis, F. Vichot, F. Wolinski, G. Paliouras, V. Karkaletsis, and C. Spyropoulos. Using machine learning to maintain rule-based named-entity recognition and classification systems. In *Proceedings of 39th Annual Meeting of the Association for Computational Linguistics*, pages 426–433, 2001.

P. Pirolli, P. Schank, M. Hearst, and C. Diehl. Scatter/gather browsing communicates the topic structure of a very large text collection. In *Conference on Human Factors in Computing Systems, CHI-96*, pages 213–220, 1996.

P. Devi Poongulhali, N. Kavitha Noel, R. Preeda Lakshmi, T. V. Geetha, and A. Manavazhahan. Tamil wordnet. In *Poceedings of the First International Conference on General WordNet*, Mysore, India, january 2002.

J. Pustejovsky. *The Generative Lexicon.* The MIT Press, Cambridge, Massachusetts, 1995.

D. Radev and W. Fan. Automatic summarization of search engine hit lists. In *Proceedings of ACL'2000 Workshop on Recent Advances in Natural Language Processing and Information Retrieval*, Hong Kong, China, 2000.

D. R. Radev. A common theory of information fusion from multiple text sources, step one: Cross-document structure. In *Proceedings of the 1st ACL SIGDIAL Workshop on Discourse and Dialogue*, pages 74–83, New Brunswick, New Jersey, 2000. Association of Computational Linguistics.

D.R. Radev, W. Fan, , and Z. Zhang. Webinessence: A personalized web-based multi-document summarization and recommendation system. In *Proceedings of NAACL Workshop on Automatic Summarization*, Pittsburgh, PA, 2001.

J. Ragetli. *Towards Concept-based Structuring of Electronic Scientific Information.* Master Thesis at the Institute for Logic, Language and Computation, University of Amsterdam, 2001.

M. Rajman and A. Bonnet. Corpora-based linguistics: new tools for natural language processing. In *1st Annual Conference of the Association for Global Strategic Information*, Germany, 1992. Bad Kreuznach.

L. A. Ramshaw and M. P. Marcus. Text chunking using transformation-based learning. In *Third ACL Workshop on Very Large Corpora*, pages 82–94. Kluwer, 1995.

A. Ratnaparkhi. *Maximum Entropy Models for Natural Language Ambiguity Resolution.* Ph.D. Dissertation. University of Pennsylvania, 1998.

D. R. Raymond and P. W. Tompa. Hypertext and the oxford english dictionary. In *Hypertext'87 Papers*, pages 143–153, The University of North Carolina, Chapel Hill, North Carolina, 1987. ACM.

D. R. Raymond and P. W. Tompa. Hypertext and the oxford english dictionary. *Communications of the ACM*, 31(7):871–879, 1988.

U. Reimer and U. Hahn. Text condensation as knowledge base abstraction. In *Proceedings of the 4th Conference on Artificial Intelligence Applications*, pages 338–344, Washington, D. C., 1988. IEEE Computer Society.

P. Resnik. *Selection and Information: A Class-Based Approach to Lexical Relationships.* Ph.D. thesis. Department of Computer and Information Science, University of Pennsylvania, 1993.

P. Resnik and D. Yarowsky. A perspective on word sense disambiguation methods and their evaluation, 1997.

P. K. Resnik. Disambiguating noun groupings with respect to wordnet senses. In *Proceedings of the Third Workshop on Very Large Corpora*, pages 54–68, Somerset, New Jersey, 1995. ACL.

P. S. Resnik. Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research*, 11:95–130, 1999.

G. Rigau. *Automatic Acquisition of Lexical Knowledge from MRDs.* PhD Thesis, Departament de Llenguatges i Sistemes Informàtics.– Universitat Politècnica de Catalunya. – Barcelona, 1998.

C. Rosé. *Facilitating the rapid development of language understanding interfaces for tutoring systems.* Technical Report FS-00-01, Papers from the AAAI Fall Symposium on Building Dialogue Systems for Tutorial Applications, 2000. URL `citeseer.nj.nec.com/rose00facilitating.html`.

A. Roventini, A. Alonge, F. Bertagna, N. Calzolari, R. Marinelli, B. Magnini, M. Speranza, and A. Zampolli. Italwordnet: A large semantic database for the automatic treatment of the italian language. In *Poceedings of the First International Conference on General WordNet*, Mysore, India, january 2002.

G. Salton. *Automatic text processing.* Addison-Wesley, 1989.

G. Salton, A. Singhal, M. Mitra, and C. Buckley. Automatic text structuring and summarisation. *Information Processing and Management*, 33(2):193–207, 1997.

G. Sampson. *English for the computer: The SUSANNE corpus and analytic scheme.* Clarendon Press, Oxford, 1995.

C. Sanrach and M. Grandbastien. ECSAIWEB: A web-based authoring system to create adaptive learning systems. In *Adaptive Hypermedia and Adaptive Web-Based Systems. Proceedings of the AH'2000 conference. LNCS 1892*, pages 214–226, Heidelberg, 2000. Springer-Verlag.

A. Setzer and R. Gaizauskas. A pilot study on annotating temporal relations in text. In *Workshop on Temporal and Spatial Information Processing, ACL-2001*, Toulouse, France, 2001.

S. Siegel and N. J. Castellan. *Nonparametric Statistics for the Behavioral Sciences.* McGraw-Hill, second edition, 1988.

SIMPLE. Semantic information for multifunctional plurilingual lexicons: An overview, 2002. URL `http://www.ub.es/gilcub/SIMPLE/reports/simple/Site_simple.htm`.

E. F. Skorochod'ko. Adaptive method of automatic abstracting and indexing. In *Information Processing 71: Proceedings of the IFIP Congress 71, C. V. Freiman (ed.)*, pages 1179–1182, Amsterdam, Holland, 1972.

S. Soderland. Learning information extraction rules for semi-structured and free text. *Machine Learning*, 34 (1–3):233–272, 1999.

K. Spark-Jones. *Synonymy and Semantic Classification.* Ph.D. thesis. University of Cambridge, England. Published in the Edinburgh Information Technology Series (EDITS), Sidney Michaelson and Yorick Wilks (eds.), Edinburgh University Press, 1986., 1964.

M. Specht and R. Oppermann. ACE - adaptive courseware environment. *The New Review of Hypermedia and Multimedia*, 4:141–161, 1998.

M. Stern. The difficulties in web-based tutoring, and some possible solutions. In *Proceedings of the workshop Intelligent Educational Systems on the World Wide Web, 8th World Conference of the AIED Society*, Kobe, Japan, 1997.

O. Stock and the Alfresco project team. Alfresco: enjoying the combination of natural language processing and hypermedia for information exploration. In *Mark T. Maybury, editor, Intelligent Multimedia Interfaces*, pages 197–224. AAAI Press/The MIT Press, 1993.

B. M. Sundheim. Overview of the third message understanding evaluation and conference. In *Proceedings of the Third Message Understanding Conference (MUC-3)*, 1991.

J. R. R. Tolkien. *The Lord of the Rings.* Allen and Unwin, 1968.

UMLS. *Unified Medical Language System. UMLS Knowledge Sources. 9th Edition.* National Library of Medicine, 1998.

T. A. van Dijk. *Recalling and Summarizing Complex Discourse*, pages 49–93. W. Burchart and K. Hulker (eds.), Text Processing. Berlin: Walter de Gruyter, 1979.

T. A. van Dijk. *News as Discourse.* Lawrence Erlbaum, Hillsdale, NJ, 1988.

J. Vassileva. A task-centred approach for user modeling in a hypermedia office documentation system. In *Adaptive Hypertext and Hypermedia, P. Brusilovsky, A. Kobsa and J. Vassileva (eds.)*, chapter 8, pages 209–247. Kluwer Academic Publishers, Dordrecht, 1998.

J. Vivaldi. *Extracción de Candidatos a Término mediante la combinación de estrategias heterogéneas.* Ph.D. thesis, Universitat Politècnica de Catalunya, 2002.

J. Vivaldi, L. Màrquez, and H. Rodríguez. Improving term extraction by system combination using boosting. In *Proceedings of the Joint ECML-PKDD'01 Conference*, Freiburg, Germany, 2001.

E. Voorhees. Using wordnet to disambiguate word senses for text retrieval. In *SIGIR-93*, 1993.

E. Voorhees. Overview of the trec-9 question answering track. In *Proceedings of the Ninth TREC Conference*, 2000.

E. M. Voorhees and D. K. Harman. *NIST Special Publication 500-250: The Tenth Text REtrieval Conference (TREC 2001).* Department of Commerce, National Institute of Standards and Technology, 2001.

P. Vossen. *EuroWordNet - A Multilingual Database with Lexical Semantic Networks.* Kluwer Academic Publishers, 1998.

J. W. Wallis and E. H. Shortliffe. Customized explanations using causal knowledge. In *B. C. Buchanan and E. H. Shortliffe (eds.), Rule-based Expert Systems: the MYCIN Experiments of the Stanford Heuristic Programming Project*, pages 371–388. Addison-Wesley Publishing Company, 1985.

G. I. Webb, J. Wells, and Z. Zheng. An experimental evaluation of integrating machine learning with knowledge acquisition. *Machine Learning*, 35:5, 1999. URL `citeseer.nj.nec.com/article/webb96experimental.html`.

G. Weber and M. Specht. User modeling and adaptive navigation support in www-based tutoring systems. In *A. Jameson, C. Paris and C. Tasso (eds.), Proceedings of the 6th International Conference on User Modeling*, pages 289–300. Wien: SpringerWienNewYork, 1997.

A. C. Welty and D. A. Ferucci. Instances and classes in software engineering. *Intelligence Magazine*, 10(2): 24–28, Summer 1999.

A. Wierzbicka. Apples are not a "kind of fruit". In *American Ethnologist*, volume 11, pages 313–328, 1984.

Y. A. Wilks. The interaction of knowledge sources in word sense disambiguation. *Computational Linguistics*, 27(3):321–350, 2001.

Y. A. Wilks, B. M. Slator, and L. M. Guthrie. *Electric words: Dictionaries, computers and meanings.* Cambridge, MA: MIT Press, 1996.

W. Woods and J. Schmolze. The kl-one family. *Computer and Mathematics with Applications*, 23(2–5): 133–177, 1992.

H. Wu and P. de Bra. Link-independent navigation support in web-based adaptive hypermedia. In *Proceedings of the Eleventh International World Wide Web Conference*, Hawaii, U.S.A., 2002.

H. Wu, P. de Bra, A. Aerts, and G. Houben. Adaptation control in adaptive hypermedia systems. In *Adaptive Hypermedia and Adative Web-based systems.*, Lecture Notes in Computer Science LNCS 1892, P. Brusilovsky, O. Stock and C. Strapparava (eds.), pages 250–259. Springer, Berlin, 2000.

F. Xia. Extracting tree adjoining grammars from bracketed corpora. In *In Fifth Natural Language Processing Pacific Rim Symposium (NLPRS-99)*, Beijing, China, 1999.

D. Yarowsky. Word-sense disambiguation using statistical models of roget's categories trained on large corpora. In *Proceedings of COLING-92*, pages 454–460, Nantes, France, 1992.

I. Zukerman and D. Litman. Natural language processing and user modeling: synergies and limitations. *User Modeling and User Adapted Interaction*, 11(1–2):129–158, 2001.

I. Zukerman and R. McConachy. Consulting a user model to address a user's inferences during content planning. *User Modeling and User Adapted Interaction*, 3(2):155–185, 1993.

Reunido el tribunal que suscribe en el día de la fecha, acordó calificar la presente Tesis doctoral con SOBRESALIENTE CUM LAUDE

Madrid, 29-5-03

ANTONIO MORENO SANDOVAL

ROBERTO MORIYON

Johann Schlichter

ESTRELLA PULIDO

ENRICO ALBERRO